

Universidad Carlos III de Madrid

Escuela Politécnica Superior



Ingeniería Técnica en Informática de Gestión

Proyecto Fin de Carrera

“Cal App”

Aplicación Generadora de Calendarios de Exámenes

Autor: Rubén Caballero Pérez
Tutora: Elena Castro Galán

Julio, 2012

Agradecimientos

En primer lugar, me gustaría dar las gracias a Elena, por ser una tutora excelente, que me ha brindado un gran apoyo y me ha motivado para conseguir mi meta en un tiempo que tenía que tenía que cumplir, y que (yo considero) ha sido un tiempo récord. También le estaré eternamente agradecido por todo lo que me ha aportado a nivel personal, porque siempre ha estado dispuesta a ayudarme y me ha dado ánimos para poder seguir avanzando, y poder cumplir mi meta, sin una tutora como ella nunca lo habría conseguido en este tiempo “límite” que tenía establecido. Muchas gracias de todo corazón.

También quiero dar las gracias a Idoia, por su amor y apoyo incondicional, y porque sé que ella ha sufrido y ha reído tanto como yo en la realización de este proyecto, gracias por aguantarme y además encima poder darme fuerzas.

Para finalizar me gustaría agradecer también a mi familia y amigos que me han dado ánimos durante la realización de este proyecto, pero especialmente a mis padres, porque ellos son los que más han sufrido por mí y por mi futuro profesional y personal, hasta el punto de preocuparse más que yo mismo.

Muchas gracias a todos, este proyecto es para vosotros.

Índice de contenido

1. INTRODUCCIÓN	1
1.1 CONTEXTO	1
1.2 PROBLEMÁTICA ACTUAL	2
1.3 SOLUCIÓN AL PROBLEMA	3
2. ESTADO DEL ARTE	5
2.1 LOS PROBLEMAS DE SCHEDULING	5
2.2 TÉCNICAS USADAS PARA RESOLVER ESTE TIPO DE PROBLEMAS	7
3. ESPECIFICACIÓN DE REQUISITOS	13
3.1 DESGLOSE DE OBJETIVOS	14
3.2 DESCRIPCIÓN DE LOS REQUISITOS	16
3.3 CASOS DE USO	18
3.4 ESTUDIO DE VIABILIDAD	20
3.5 RECURSOS PARA LA CONSTRUCCIÓN Y FUNCIONAMIENTO DEL SISTEMA	25
4. GESTIÓN DEL PROYECTO	27
4.1 CICLO DE VIDA	28
4.2 PLANIFICACIÓN Y PRESUPUESTOS	32
5. ANÁLISIS Y DISEÑO DE LA APLICACIÓN	37
5.1 NECESIDADES DE LA INFORMACIÓN	38
5.2 ANÁLISIS FUNCIONAL	46
5.3 DISEÑO DE INTERFACES	49
6. IMPLEMENTACIÓN DE LA APLICACIÓN	54
6.1 TECNOLOGÍAS Y HERRAMIENTAS UTILIZADAS EN LA APLICACIÓN	55
6.2 IMPLEMENTACIÓN DE LAS FUNCIONALIDADES	62
6.3 PROCESO DEL ALGORITMO UTILIZADO	74
7. VALIDACIÓN Y PRUEBAS	79
7.1 PRUEBAS DE USUARIO	80
7.2 PRUEBAS DE RENDIMIENTO	85
8. CONCLUSIONES Y LÍNEAS FUTURAS	95
8.1 CONCLUSIONES	95
8.2 LÍNEAS FUTURAS	97
BIBLIOGRAFÍA	100
ANEXO A. LISTADO DE REQUISITOS	104
A.1 REQUISITOS DE USUARIO	105
A.2 REQUISITOS DE SOFTWARE	122
ANEXO B. LISTADO DE CASOS DE USO	134
ANEXO C. MANUAL DE INSTALACIÓN	141
C.1 INSTALACIÓN DE .NET FRAMEWORK 4	142
C.2 INSTALACIÓN DE SQL SERVER 2008 EXPRESS	143
C.3 INSTALACIÓN DE LA BASE DE DATOS EN SQL SERVER	150
ANEXO D. MANUAL DE USUARIO	153
D.1 PRIMERA EJECUCIÓN	156
D.2 PRIMER PASO: ASIGNATURAS	157
D.3 SEGUNDO PASO: FECHAS DE EXÁMENES	163
D.4 TERCER PASO: ALGORITMO A UTILIZAR	165
D.5 CUARTO PASO: EXPORTAR RESULTADOS	168

Índice de ilustraciones

ILUSTRACIÓN 1: DIAGRAMA DE CASOS DE USO.....	19
ILUSTRACIÓN 2: CICLO DE VIDA EN CASCADA CON RETROALIMENTACIÓN	31
ILUSTRACIÓN 3: LISTADO DE TAREAS	33
ILUSTRACIÓN 4: DIAGRAMA DE GANTT	34
ILUSTRACIÓN 5: ASIGNACIÓN DE COSTES	35
ILUSTRACIÓN 6: MODELO E/R	41
ILUSTRACIÓN 7: MODELO DE TRES CAPAS	47
ILUSTRACIÓN 8: DIAGRAMA DE COMPONENTES	48
ILUSTRACIÓN 9: DIAGRAMA DE CLASES	48
ILUSTRACIÓN 10: FLUJO ENTRE PANTALLAS	50
ILUSTRACIÓN 11: FRANJA DE LA VENTANA PRINCIPAL	51
ILUSTRACIÓN 12: BARRA DE TÍTULO DE LA APLICACIÓN.....	51
ILUSTRACIÓN 13: APLICACIÓN EN LA BARRA DE TAREAS	51
ILUSTRACIÓN 14: DISEÑO DE LAS PESTAÑAS (PASOS) DE LA PANTALLA PRINCIPAL.....	52
ILUSTRACIÓN 15: DISEÑO DEL BOTÓN PARA IMPORTAR LAS ASIGNATURAS DEL FICHERO DE ENTRADA	52
ILUSTRACIÓN 16: EJEMPLO DE GLOBO DE AYUDA PARA EL USUARIO.....	52
ILUSTRACIÓN 17: ARQUITECTURA PRINCIPAL DE LA PLATAFORMA .NET	56
ILUSTRACIÓN 18: ARQUITECTURA DE LA TECNOLOGÍA LINQ.....	58
ILUSTRACIÓN 19: TABLA “AÑO”	62
ILUSTRACIÓN 20: TABLA “ASIGNATURA”	63
ILUSTRACIÓN 21: TABLA “CUATRIMESTRE”	64
ILUSTRACIÓN 22: TABLA “CURSO”	64
ILUSTRACIÓN 23: TABLA “TITULACIÓN”	64
ILUSTRACIÓN 24: PLANTILLA DE PRUEBA DE USUARIO	79
ILUSTRACIÓN 25: TIEMPOS GENERACIÓN PRIMER CUATRIMESTRE	86
ILUSTRACIÓN 26: TIEMPOS GENERACIÓN SEGUNDO CUATRIMESTRE	87
ILUSTRACIÓN 27: TIEMPOS GENERACIÓN CONVOCATORIA EXTRAORDINARIA	88
ILUSTRACIÓN 28: PLANTILLA PARA LAS PRUEBAS DE LA EFICACIA DE LOS CALENDARIOS.....	90
ILUSTRACIÓN 29: PLANTILLA DE REQUISITO.....	104
ILUSTRACIÓN 30: PLANTILLA DE CASO DE USO	134
ILUSTRACIÓN 31: PÁGINA DE DESCARGA DE .NET FRAMEWORK 4.....	142
ILUSTRACIÓN 32: INSTALANDO NET FRAMEWORK 4	142
ILUSTRACIÓN 33: DESCARGA DE SQL SERVER 2008 EXPRESS	143
ILUSTRACIÓN 34: INSTALACIÓN DE SQL SERVER 2008 EXPRESS (1)	144
ILUSTRACIÓN 35: INSTALACIÓN DE SQL SERVER 2008 EXPRESS (2)	145
ILUSTRACIÓN 36: INSTALACIÓN DE SQL SERVER 2008 EXPRESS (3)	145
ILUSTRACIÓN 37: INSTALACIÓN DE SQL SERVER 2008 EXPRESS (4)	146
ILUSTRACIÓN 38: INSTALACIÓN DE SQL SERVER 2008 EXPRESS (5)	147
ILUSTRACIÓN 39: INSTALACIÓN DE SQL SERVER 2008 EXPRESS (6)	148
ILUSTRACIÓN 40: INSTALACIÓN DE SQL SERVER 2008 EXPRESS (7)	149
ILUSTRACIÓN 41: INSTALACIÓN DE LA BASE DE DATOS.....	150
ILUSTRACIÓN 42: INSTALACIÓN DE LA BASE DE DATOS.....	151
ILUSTRACIÓN 43: MENSAJE DE INFORMACIÓN SOBRE LA APLICACIÓN	154
ILUSTRACIÓN 44: MENSAJE DE AVISO INDICANDO QUE NO HAY CONEXIÓN	155
ILUSTRACIÓN 45: VENTANA DE SERVICIOS DE WINDOWS	155
ILUSTRACIÓN 46: MENSAJE INDICANDO QUE NO HAY DATOS DE ASIGNATURAS RELATIVOS AL AÑO ACTUAL	156
ILUSTRACIÓN 47: PANTALLA PRINCIPAL CUANDO NO HAY ASIGNATURAS (PRIMER PASO).....	156
ILUSTRACIÓN 48: MENSAJE INDICANDO QUE YA HAY DATOS.....	157
ILUSTRACIÓN 49: CUADRO DE DIÁLOGO PARA ABRIR NUESTRO FICHERO EXCEL	158
ILUSTRACIÓN 50: MENSAJE DE ESPERA	158
ILUSTRACIÓN 51: MENSAJE DE ÉXITO AL LEER FICHERO DE ENTRADA	158
ILUSTRACIÓN 52: PANTALLA DE ELECCIÓN DE OPTATIVAS	159

ILUSTRACIÓN 53: PANTALLA DE VISUALIZACIÓN DE OPTATIVAS.....	160
ILUSTRACIÓN 54: PANTALLA DE VISUALIZACIÓN DE ASIGNATURAS DEL GRADO EN INFORMÁTICA	161
ILUSTRACIÓN 55: PANTALLA DE VISUALIZACIÓN DE ASIGNATURAS DEL DOBLE GRADO	162
ILUSTRACIÓN 56: PANTALLA PRINCIPAL DE LA APLICACIÓN (SEGUNDO PASO).....	163
ILUSTRACIÓN 57: SELECCIÓN DEL PERÍODO	163
ILUSTRACIÓN 58: SELECCIÓN DE FECHAS DE INICIO Y FIN DEL PERÍODO.....	164
ILUSTRACIÓN 59: NO CONTAR CON LOS SÁBADOS.....	164
ILUSTRACIÓN 60: PANTALLA PRINCIPAL (TERCER PASO)	165
ILUSTRACIÓN 61: SELECCIÓN DE ALGORITMO	165
ILUSTRACIÓN 62: PANTALLA PRINCIPAL DURANTE EL PROCESO DE GENERACIÓN	166
ILUSTRACIÓN 63: MENSAJE DE TERMINACIÓN DE LA GENERACIÓN DEL CALENDARIO.....	166
ILUSTRACIÓN 64: VENTANA DE PREVISUALIZACIÓN DEL CALENDARIO	167
ILUSTRACIÓN 65: PANTALLA PRINCIPAL (CUARTO PASO).....	168
ILUSTRACIÓN 66: MENSAJE DE ÉXITO AL EXPORTAR	168

1. Introducción

1.1 Contexto

La tarea de planificar exámenes para una titulación de la Universidad es un trabajo complicado, al que tienen que enfrentarse los profesores año tras año. Es bastante tedioso el tener que asignar los exámenes de las asignaturas de una titulación de forma que el mayor número de alumnos quede satisfecho con el resultado, pero... ¿Por qué es tan complicado?

Hay que tener en cuenta que una titulación (Grados) tiene cuatro cursos, y en determinadas titulaciones (Dobles Grados) tienen hasta seis cursos. Cada curso tiene un número de asignaturas distintas divididas en el primer y segundo cuatrimestre.

En cuanto a los períodos de exámenes, hay que tener en cuenta que existe una convocatoria por cuatrimestre, además de la convocatoria extraordinaria. Es decir existen tres convocatorias de exámenes donde los profesores tienen que asignar esos exámenes de las asignaturas. La convocatoria del primer cuatrimestre es en el mes de enero, la del segundo cuatrimestre es en mayo, y la extraordinaria en junio.

Lo que hace realmente complicado este trabajo de asignar los exámenes es que para una titulación, en un cuatrimestre, existe un gran número de asignaturas de diferentes cursos a asignar, pero sólo se pueden asignar en un rango de días que es muy corto para la cantidad de asignaturas que existen, y la solución elegida tiene que ser la que mejor se adapte a los alumnos.

Esta solución que se tiene que adaptar lo mejor posible a las necesidades de los alumnos debe cumplir una serie de restricciones en el mayor grado de lo posible y consumiendo el menor tiempo disponible.

Actualmente, los profesores realizan este trabajo manualmente, consumiendo gran parte de su tiempo y quebraderos de cabeza innecesarios cuando asignan los exámenes de una convocatoria intentando que sea de la mejor manera que encuentran.

Y si además se trata de la convocatoria extraordinaria, donde se tiene en cuenta la totalidad de las asignaturas de la titulación, y tenemos el mismo número de días disponible, el trabajo se complica aún más, ya que se solaparán muchísimas más asignaturas.

1.2 Problemática actual

Actualmente, este rango de días de una convocatoria de exámenes consta de catorce días naturales, pero de estos días hay que excluir los domingos, ya que no puede haber exámenes un domingo. Así que existen doce días para asignar todas las asignaturas de una convocatoria donde cada día tiene dos turnos, el turno de mañana y el turno de tarde y las asignaturas se asignarán en esos turnos, pero los sábados solo tienen disponible el turno de mañana, ya que no podrá haber exámenes un sábado por la tarde.

Por lo tanto, para entender la dificultad a la que se enfrentan los profesores actualmente, vemos que tenemos veintidós turnos para asignar un gran número de asignaturas de cursos distintos y que esta asignación hay que hacerla de la manera más óptima posible para que la mayoría de alumnos pueda ir a todos sus exámenes sin tener que solicitar un cambio de fecha porque le coinciden dos exámenes, o que deje una asignatura por ir a otra cuyo examen es el mismo día, por ejemplo. Lo que se busca es reducir este tipo de casos en la manera de lo posible.

A esto hay que añadirle que existe la posibilidad de que en un futuro los sábados tampoco se tengan en cuenta en las convocatorias de exámenes, lo que reduce el número de turnos disponibles para asignar exámenes a veinte turnos (diez días). Los alumnos agradecerían el no poder tener un examen un sábado, pero esto también hace que la dificultad a la hora de asignar los días para los exámenes sea mayor, ya que tenemos menos días.

Por último y no menos importante, tenemos otra dificultad añadida. Hay titulaciones que comparten asignaturas de dos titulaciones distintas, hablamos de los Dobles Grados. Para asignar los exámenes de estas titulaciones se tienen que tener en cuenta las dos titulaciones de las que están compuestas, ya que las asignaturas de los Dobles Grados son un subconjunto de las que titulaciones de las que están compuestas.

Este proyecto está enfocado en uno de esos casos, en concreto, el de asignar los exámenes para la titulación “Doble Grado en Informática – ADE” y la titulación “Informática”. Hay que tener en cuenta que estas titulaciones también disponen de “especialidades” a partir de determinados cursos. Esto supone que los alumnos que elijan una especialidad solo tendrán asignaturas de esa especialidad y no de las demás, por lo tanto también habrá que tener eso en cuenta a la hora de asignar los exámenes de estas titulaciones, además de los cursos.

1.3 Solución al problema

Para solucionar este problema se ha optado por el desarrollo de una aplicación de escritorio que permita hacer estas tareas de forma automática, eliminando una carga de trabajo a los profesores que puede ser innecesaria, además de ahorrarse todo ese tiempo de planificación.

Esta aplicación deberá ser capaz de generar calendarios de exámenes para estas titulaciones (en concreto, el “Doble Grado en Informática – ADE” e “Informática”). Estas dos titulaciones son algo peculiares respecto a las demás, ya que tienen especialidades y también asignaturas compartidas, cosa que la mayoría de las titulaciones no tienen. Por eso la aplicación está enfocada a estas dos titulaciones, aunque se ha desarrollado de la manera más abierta posible para poder introducir en un tiempo futuro la posibilidad de implementarla para que acepte todas las titulaciones de la Universidad.

Esta aplicación será capaz de procesar una entrada con los datos de todas las asignaturas de estas titulaciones correctamente, y generar exámenes para las mismas en un formato amigable y legible para el profesor, separados por curso si se desea y por titulación.

Como ya se ha comentado anteriormente, para la asignación de exámenes hay que cumplir una serie de restricciones para que el calendario sea lo más “humano” posible desde el punto de vista de los alumnos.

De todas estas restricciones, que se enumerarán más adelante, se encargará el algoritmo elegido para procesarlas y de cumplirlas, siempre en el mayor grado de lo posible, ya que en la mayoría de los casos, será imposible cumplir todas las restricciones. Este algoritmo tratará de encajar los exámenes para que cuadren de la manera más óptima posible y en el menor tiempo posible.

En términos generales, esta aplicación será lo más legible, simple e intuitiva que se pueda para que su manejo sea más fácil y más amigable para el usuario, además de disponer de suficientes opciones para cumplir los distintos requisitos, tanto a nivel del calendario, como de la aplicación.

2. Estado del arte

2.1 Los problemas de Scheduling”

En este apartado se tratará de explicar la necesidad de las aplicaciones dedicadas a resolver problemas de planificación (“Scheduling”), y también de explicar distintos métodos que se han ido llevando a cabo hasta la actualidad para resolver estos problemas de la manera más eficiente posible, en función de los resultados obtenidos.

En realidad, estos problemas de planificación se dan en una gran variedad de casos distintos y el ser humano ha tenido y tiene que hacer frente a este tipo de problemas que se le plantean, pero ¿en qué consisten estos problemas?

Los problemas de Scheduling parten siempre de situaciones similares. Tenemos un rango de tiempo que es finito, aunque podría ser otro recurso que no fuera el tiempo, y donde tenemos que planificar una serie de tareas que deben cumplir un conjunto de restricciones compitiendo por esos recursos, donde cada restricción tiene distinto peso, que debe ser tenido en cuenta.

Estos problemas se suelen designar como problemas de optimización combinatoria porque dadas las condiciones anteriores, tendremos un enorme número de combinaciones donde nos tenemos que quedar con la más óptima que se encuentre. Aquí la solución óptima es relativa, ya que podemos definirla en función del tiempo, o en función de los resultados, o de los recursos empleados, o una combinación de varias. Ahora enfocaré el problema aplicado al proyecto, para entenderlo mejor. Si nos sumergimos más en la naturaleza del problema, vemos que también está clasificado como un problema de “Timetabling”, o planificación de horarios, que en realidad es un subtipo de los problemas de Scheduling, pero la diferencia está en que el recurso finito donde tenemos que planificar nuestras tareas es el tiempo.

El ejemplo típico de un problema de Timetabling, es el de resolver una planificación de horarios de un centro de enseñanza por ejemplo, donde también hay que cumplir en el grado de lo posible una gran cantidad de restricciones como las aulas disponibles, los grupos compartidos, los profesores, etc. para asignar asignaturas unas horas a la semana.

En nuestro problema tenemos muchas restricciones, como veremos después, una gran cantidad de “tareas” a asignar, que serán los exámenes de las asignaturas, y un tiempo finito en el que asignarlas, que será la franja de doce días de la que consta una convocatoria de exámenes.

Debemos planificar en ese tiempo todos los exámenes de las asignaturas de esa convocatoria respetando esas restricciones de la manera más óptima para obtener un resultado de ese problema de Timetabling que sea óptimo y que sea lo mejor posible para que el menor número de alumnos quede descontento con él o tenga que alterarlo cambiando algún examen, lo que también supondría un problema para los profesores.

Existe un considerable cuerpo de desarrollo en el campo de la elaboración automática de horarios (Burke, Elliman & Weare 1996). Durante los últimos años se ha investigado mucho en este ámbito y ha sido el origen de un importante grupo de trabajo WATT, (Working Group on Automated Timetabling) y de una conferencia periódica asociada a dicho grupo de trabajo PATAT, (Practice and Theory of Automated Timetabling).

Durante estos años se han aplicado muchas y diversas técnicas en el problema de la elaboración automática de horarios (que es un problema del mismo tipo al que nos enfrentamos en este proyecto con un calendario de exámenes): coloreado de grafos, programación lógica con restricciones, algoritmos genéticos y programación evolutiva. Varios de estos métodos han producido buenos resultados y algunos se han empleado en varias instituciones en todo el mundo.

2.2 Técnicas usadas para resolver este tipo de problemas

Ya hemos visto los ingredientes necesarios para obtener un problema de esta índole, cuya solución no es para nada trivial, y requiere de métodos eficientes para resolverlo. En este apartado veremos distintos métodos que el ser humano ha ido desarrollando para resolver estos problemas de la mejor manera posible, optimizando los resultados y los recursos con los que se dispone.

Lo realmente interesante de la resolución de este tipo de problemas es el algoritmo que se utiliza para ir asignando las tareas y cumplir siempre que se pueda, las restricciones que se imponen en el problema.

Como ya se ha comentado, para resolver estos problemas necesitamos algoritmos inteligentes para encontrar las mejores soluciones optimizando nuestros recursos. Es por esto por lo que el ser humano ha desarrollado y utilizado algoritmos de búsqueda heurística propios del área de la Inteligencia Artificial para resolver este tipo de problemas, como el A*, o el IDA* o los algoritmos en Escalada, o la línea de investigación que surgió con los algoritmos genéticos, los cuales han sido objeto y son de mucha investigación y se ha demostrado su gran eficacia para resolver este tipo de problemas, aparte de muchos otros.

Como estos problemas también son de optimización se pueden enfocar también hacia algoritmos propios de la Investigación Operativa, como la coloración de grafos, o el problema del Cubrimiento. No hay que olvidar que la rama de la Investigación Operativa es una rama de las matemáticas, dedicada a optimizar (ya sea maximizar beneficios o minimizar costes) el funcionamiento de sistemas con gran cantidad de tareas y recursos limitados, por eso también se pueden encontrar soluciones a este tipo de problemas.

A continuación explicaremos como funcionan estos algoritmos y las distintas aplicaciones que tienen en la vida real, en base a los artículos y libros consultados. Voy a separar los algoritmos por sus categorías:

❖ **Algoritmos basados en técnicas de Búsqueda Heurística:** Estas técnicas están englobadas dentro de la inmensa rama de los problemas de búsqueda de la Inteligencia Artificial. La búsqueda heurística es una técnica que permite encontrar una solución en un problema eliminando caminos que pensamos que no nos llevarán a ninguna solución.

Por eso, se habla de búsqueda informada, porque tenemos un conocimiento parcial sobre el dominio y mediante este conocimiento podremos ir por el camino que creemos más adecuado en cada paso, desechando otras soluciones. La manera de hacer esto es mediante una función de evaluación que en cada paso nos de unos valores cualitativos en cada nodo que exploremos para los siguientes nodos a explorar, eligiendo así el mejor sucesor que encontremos para explorarlo, este es el conocimiento parcial que tenemos en el problema.

Los métodos de búsqueda no informada se desechan por completo para la posible solución a este problema, ya que son muy ineficientes en la mayoría de los casos y se siguen estrategias de anchura o profundidad que no tienen en cuenta la calidad de los estados, por lo que serían más costosos y menos óptimos.

Estos métodos están orientados a reducir la cantidad de búsqueda requerida para encontrar una solución, conociendo esa calidad de los posibles estados a explorar.

Dentro de este tipo de técnicas se ha utilizado más de una para resolver este tipo de problemas, incluso muchas veces combinando más de una en un mismo problema para obtener mejores algoritmos y, por tanto, mejores soluciones.

Se ha demostrado en muchos trabajos y artículos que estas técnicas han obtenido resultados muy satisfactorios, sobretodo utilizando metaheurísticas y métodos híbridos, las cuales se han utilizado en su mayoría en estos problemas de optimización combinatoria.

Dentro de los ejemplos más significativos de estas técnicas y donde se ha investigado mucho son la Búsqueda Tabú, que es un tipo de búsqueda local, pero utiliza estructuras de memoria para aumentar el rendimiento; la técnica basada en colonias de hormigas, la cual se basa en una representación abstracta de el comportamiento de las hormigas donde cada individuo se comunica con otro por medio de feromonas; la búsqueda Best-First (Primero el mejor), donde también hay muchas técnicas de resolución, como el A* que es un algoritmo de búsqueda heurística bastante eficiente en la mayoría de los casos; los algoritmos voraces, los cuales siguen una heurística fija en elegir la opción más óptima en cada paso local con la esperanza de obtener una solución óptima; los algoritmos en escalada, etc.

- ❖ **Algoritmos genéticos:** Este tipo de algoritmos constituye una de las bases de la investigación actual en este tipo de problemas de Timetabling, por la eficacia demostrada y por la evolución que han tenido, ya que antes se utilizaban para otro tipo de problemas de optimización y es actualmente cuando se han empezado a utilizar para problemas de generación de horarios, obteniendo bastante éxito por sus soluciones óptimas en cuanto a los resultados obtenidos, por eso voy a explicarlo un poco más.

Estos métodos están basados en el proceso genético de los organismos vivos. Una vez más, vemos como la abstracción de situaciones naturales aplicados a la Informática dan sus frutos. Trabajan con una población de individuos y cada uno de estos individuos representa una solución factible al problema, esto es un valor que tiene cada uno y que equivaldría al grado de efectividad que tiene el organismo para competir por unos recursos, como vemos la analogía es clara.

Cuanto mayor sea este valor más se adapta el individuo al problema, lo que significa que será mayor la probabilidad de que ese individuo se reproduzca, cruzando su material genético con otro individuo seleccionado de igual forma.

Este cruce producirá nuevos individuos descendientes de los anteriores, los cuales comparten algunas de las características de sus padres.

Así produciremos una nueva población de posibles soluciones, la cual contiene una mayor proporción de buenas características en comparación con la anterior. De esta manera van siendo exploradas las áreas más prometedoras del espacio de búsqueda.

Es posible que este algoritmo no encuentre una solución óptima al problema, pero si ha sido bien diseñado es muy probable de que sí sea así.

- ❖ **Algoritmos propios de la Programación Lineal:** También se ha realizado investigación con estos métodos para resolver este tipo de problemas, como el método del “Branch and Bound” (Ramificación y acotación), que también se ha investigado su eficacia y se ha comprobado que se puede utilizar para resolver problemas de optimización.

En este método se genera un árbol de nodos, que son soluciones, pero la particularidad de este método es que además de explorar el árbol, el algoritmo se encarga de detectar en que ramificación las soluciones ya no son óptimas, y si ocurre esto, podar esa rama del árbol para no continuar por caminos inútiles y malgastar recursos.

La coloración de grafos es otra técnica que se engloba dentro de los algoritmos de la Investigación Operativa, y que consiste en buscar la menor cantidad posible de colores para colorear un grafo, de tal forma que los nodos adyacentes no pueden ser del mismo color. Esto se puede aplicar también a problemas de optimización y se han realizado numerosos trabajos sobre ello.

Como conclusión de este apartado he de decir que en la inmensa mayoría de los trabajos y artículos de investigación que he consultado las técnicas que se han utilizado han resultado ser técnicas híbridas, combinando diferentes técnicas entre sí, para intentar obtener las mejores soluciones al problema, y probando la eficacia de las mismas cuando se unen.

Anteriormente ya se ha probado la técnica de soluciones basadas en el conocimiento para problemas en los que planificación con recursos limitados (Meisels, Kuflink & Gudes 1991), diseño de planes de estudios (Khoo & Pemberton 1993), calendarios laborables (Meisels, Gudes & Solotorevsky 1995), elaboración automática de horarios escolares (Khader & Buchanan 1995), o elaboración automática de horarios universitarios (Chow, Loo, Kwan & Martinsons 1991).

En este campo hay miles de publicaciones al respecto, y cada una propone maneras distintas de resolver este tipo de problemas, ya que se pueden hacer de multitud de maneras distintas.

Es importante destacar que la mayoría de investigaciones están enfocadas a problemas más específicos, y no a un problema genérico. Actualmente cada vez más se investiga y se buscan soluciones genéricas para todos los problemas de este tipo, el problema es que las mejores soluciones se consiguen con implementaciones más específicas. El objetivo de las líneas de investigación actualmente es explorar planteamientos que permitan operar a un nivel de generalidad más alto de lo que es posible actualmente (Burke, et. al., 2000), (Burke & Petrovic, 2002).

3. Especificación de Requisitos

En este apartado se enumerarán y explicarán todos los requisitos de cualquier tipo de los que consta la aplicación. También se expondrán los objetivos a cumplir por la aplicación, así como los casos de uso de la misma, y se hará un enfoque al estudio de viabilidad del sistema.

Para poder llevar a cabo la identificación de los requisitos de la aplicación, hay que extraer estos requisitos del cliente, viendo al cliente como la persona que solicita la implementación del aplicativo. En realidad, el cliente será cualquier profesor que vaya a utilizar la aplicación, pero la persona que me ha planteado el problema a resolver es mi tutora, Elena Castro Galán.

Se han desarrollado numerosas técnicas para la extracción de requisitos, pero una de las técnicas que mejor funciona es la de mantener constantes entrevistas cara a cara con el cliente, para extraer todos los requisitos que tiene en mente y no hacerse ideas equivocadas sobre los objetivos a cumplir. Esta es una forma eficiente para extraer estos requisitos.

Para esta finalidad se han llevado a cabo varias entrevistas, para extraer las funcionalidades de forma creciente, es decir, en la primera entrevista se expuso una idea general del problema que se quería resolver, con los objetivos generales que había que cumplir, y en las sucesoras se ha ido profundizando más en el “cómo”, extrayendo nuevos requisitos para implementar, y formando mejor la idea de la aplicación y de sus casos de uso.

Se ha pasado por el siguiente proceso para llegar a extraer los requisitos de la aplicación:

1. A partir de las entrevistas mantenidas con el cliente, se extraen una serie de Requisitos de Usuario.
2. También partiendo de estas entrevistas y de los requisitos encontrados se van formando los distintos casos de uso a implementar por la aplicación, que son las posibles acciones que el usuario puede hacer durante la ejecución del programa.
3. A partir de todos los requisitos de usuario que hemos obtenido se procede a identificar los Requisitos Software, definiendo funcionales y no funcionales.

Este proceso es cíclico, es decir, se repetirá con cada entrevista mantenida con el cliente.

3.1 Desglose de objetivos

Los objetivos generales de la aplicación se obtienen extrayendo la información que nos proporciona el cliente, el cual nos dice qué es lo que quiere que haga el sistema, qué funcionalidades debe tener.

Para esto, nos hemos valido de la entrevista personal con el cliente como ya se ha comentado. Se ha optado por el análisis de protocolo, el cual parte de la idea de que el cliente ya sabe que es lo que quiere que haga la aplicación, y en base a esto se obtienen los requisitos.

Esta técnica se ha utilizado en el problema porque el cliente ya tenía una idea lo suficientemente clara de las funcionalidades básicas que tenía que cubrir la aplicación, por lo tanto se ha elegido adecuadamente a las circunstancias.

En las entrevistas mantenidas con el cliente se vieron los siguientes puntos: Se partió de una necesidad vigente que tienen los profesores, y es la de poder generar calendarios de exámenes de forma automática, ya que hasta ahora se han ido realizando de manera manual, lo cual es muy costoso e innecesario. Partiendo de esa necesidad el cliente me propuso el realizar una aplicación que se encargara de hacer este trabajo de forma automática y que fuera lo más óptimo posible.

El enfoque de la aplicación fue el de generar un calendario de exámenes válido para las titulaciones “Doble Grado en Informática – ADE” y “Grado en Ingeniería Informática”, entonces se vio que en los horarios de los dos planes de estudios se comparten grupos en muchas asignaturas. Estudiando los dos planes de estudio, pude comprobar que todas (en principio) las asignaturas relativas a la rama informática del Doble Grado compartían grupo con sus equivalentes en el Grado de Ingeniería Informática, llegando a la conclusión de que estas asignaturas debían compartir el mismo examen en el mismo día.

Por tanto, con las reuniones mantenidas se identificaron una serie de objetivos generales que debía cumplir la aplicación, y que son los siguientes:

- Implementar una aplicación que permita generar los calendarios de exámenes del primer cuatrimestre, el segundo, y la convocatoria extraordinaria, de las dos titulaciones “Doble Grado en Informática – ADE” y “Grado en Ingeniería Informática”.

- Almacenar de forma persistente los datos de las asignaturas, así como poder cargarlos en cualquier momento si se desea, ya que cada año variarán estos datos.
- Incluir también las asignaturas de la rama de ADE, cuyas fechas de exámenes ya vendrán dadas, ya que no dependen de nosotros, sino de los profesores de ADE.
- Tener la opción de elegir un subconjunto de asignaturas optativas, si no se van a impartir todas.
- Los calendarios generados deben poder guardarse y deben tener un formato adecuado y presentable, con un buen diseño, para que el profesor pueda consultarlos cómodamente.

3.2 Descripción de los requisitos

En este apartado se expone cada requisito de usuario de la aplicación, así como una descripción de lo que significa cada requisito. De esta manera, sabemos qué requisitos debe cumplir la aplicación en cuanto a todo el proceso de generación del calendario.

Aquí no se hace una definición técnica con todas las tablas de requisitos, sino que se explica en qué consiste cada requisito que el cliente nos ha solicitado.

El listado formal de los requisitos puede encontrarse en el “Anexo B: Listado de requisitos”.

Descripción de los requisitos de usuario:

- Debe existir la posibilidad de dar a elegir al usuario la fecha de inicio y la fecha de fin del calendario, aunque siempre sea de doce días, porque puede ser que en el futuro no sea así.
- Los domingos no se deben contar en la franja de días del calendario de exámenes, ya que ese día no puede haber ningún examen.
- Cada día debe tener dos turnos para colocar los exámenes, el turno de mañana y el turno de tarde.
- Los sábados solo deben tener un turno: el turno de mañana, ya que un sábado por la tarde no podrá haber ningún examen.
- Debe existir la posibilidad de dar a elegir al usuario si quiere que se cuenten los sábados para el calendario de exámenes o no, ya que probablemente en un futuro próximo tampoco podrá haber exámenes los sábados.
- Debe existir la posibilidad de que se dé a elegir al usuario la opción de cargar un fichero de entrada con los datos de las asignaturas del curso actual, ya que estos datos variarán cada año.
- El fichero de entrada tendrá un formato específico y deberá contener todos los datos relevantes de las asignaturas relativos a las dos titulaciones.
- El fichero de entrada no deberá admitir valores que no sean válidos para los diferentes datos de las asignaturas.
- Se deberá dar la opción al usuario de elegir qué asignaturas optativas de la titulación Grado en Ingeniería Informática se van a impartir este año, ya que no siempre se impartirán todas.

- Debe de existir un almacenamiento persistente para guardar los datos de las asignaturas del curso actual, para no tener que cargar el fichero de entrada cada vez que se ejecute la aplicación.
- Las asignaturas deberán tener un dato que nos diga si es una asignatura difícil o no, porque es un dato que tendrá que ser tratado en la generación del calendario.
- Deberá implementarse un algoritmo óptimo para colocar los distintos exámenes de las asignaturas en los días que tenemos disponibles, cumpliendo en el mayor grado de lo posible las restricciones que nos vienen impuestas en la generación.
- Las salidas de nuestra aplicación, es decir, los calendarios de exámenes, deberán estar en un formato que sea simple, legible y elegante para el usuario, mostrando los datos relevantes de las asignaturas.

A continuación se describen las restricciones para la generación del calendario (que son implementadas en el algoritmo), ya que, aunque sean requisitos de usuario, no se tienen por qué cumplir obligatoriamente, ya que sería imposible cumplir todas, pero si en el mayor grado posible:

- En días adyacentes del calendario no debería haber exámenes de asignaturas que sean del mismo curso y especialidad.
- No debería haber exámenes de asignaturas del mismo curso y misma especialidad en el mismo día y turno del calendario.
- En el mismo día y en el mismo turno no debería haber exámenes de asignaturas que sean de cursos consecutivos y misma especialidad.
- Sería deseable que no haya exámenes de asignaturas de cursos consecutivos y misma especialidad en el mismo día pero distinto turno del calendario.
- Sería muy deseable que no haya exámenes de asignaturas del mismo curso y especialidad en el mismo día pero en distinto turno.
- Las asignaturas con dificultad alta deberán estar separadas lo máximo posible unas de otras.
- Es preferible que los exámenes sean colocados en el turno de mañana, siempre que sea posible.

Sobre estas últimas restricciones, que también son requisitos de usuario, se hablará más en el sexto apartado, y sobre el “daño” que hace cada una cuando no se cumple.

3.3 Casos de uso

En este apartado definiremos lo que es un caso de uso, para que se utiliza y como se representa.

Un caso de uso se puede definir como un “texto narrativo” que describe el uso que hace un actor del sistema, en este caso, la aplicación. Un actor puede ser humano o ser otro sistema externo, en nuestro será un usuario humano.

También se puede definir un caso de uso como una forma que tiene el usuario de utilizar el sistema. Un sistema estará compuesto típicamente de varios casos de uso, que representan distintas funcionalidades a nivel de usuario que pueden realizarse sobre ese sistema y sus actores, en respuesta a un evento que desencadena el actor principal sobre el sistema.

Aunque los casos de uso no sean exactamente requisitos, ilustran requisitos en sus representaciones. Un caso de uso debe describir un camino sin bifurcaciones, por eso cada camino alternativo en una funcionalidad será un caso de uso distinto.

Los casos de uso se extraen de los requisitos de usuario, y un caso de uso puede satisfacer uno o varios de estos requisitos, en concreto aquellos que estén directamente involucrados con la actividad que resuelve el caso de uso.

Lo principal a tener en cuenta es que un caso de uso no debe describir el funcionamiento interno del sistema, sino qué es lo que hace el sistema desde el punto de vista del usuario de una forma abstracta.

Para representar estos casos de uso lo haremos mediante la notación UML (Unified Modeling Language, Lenguaje de modelado unificado), ya que es una manera estándar de especificar y documentar un sistema de software. Para representarlo de esta manera dibujaremos el Diagrama de Casos de Uso, que contendrá los siguientes elementos:

- Actor: Se representa con una figura de persona.
- Caso de uso: Se representa mediante una elipse con el nombre del caso de uso contenido en ella.
- Asociación: Se representa mediante una línea, y muestra las relaciones entre actores y casos de uso.

Diagrama de Casos de Uso:

En esta página se muestra el diagrama de casos de uso de la aplicación usando el lenguaje UML, como se explica en la página anterior:



Ilustración 1: Diagrama de Casos de Uso

3.4 Estudio de viabilidad

Como ya hemos definido los requisitos y los casos de uso del sistema ahora procederemos a redactar un estudio de la viabilidad del software, explicando por qué se han utilizado unas tecnologías y no otras, para alcanzar la solución propuesta.

La solución se implementará mediante una aplicación para PC monousuario, como ya hemos dicho, cuya finalidad será generar calendarios de exámenes para las titulaciones Grado en Ingeniería Informática y Doble Grado en Informática – ADE.

El sistema estará formado por esta aplicación gráfica, más una base de datos donde se guardaran todos los datos de las asignaturas del curso actual.

Según la especificación de nuestro sistema se ha utilizado una arquitectura de aplicación de tres capas: Modelo – Vista – Controlador, para desarrollarlo, del cual hablaremos más adelante en el sexto apartado.

Para realizar este estudio habrá que hacer varias decisiones, comparando diferentes alternativas, para quedarnos con la que mejor se adapte a los requerimientos del cliente.

Habrá que decidir:

- El sistema operativo en el que vamos a implantar la aplicación.
- El lenguaje de programación que usaremos para su desarrollo.
- El Sistema Gestor de la Base de Datos.

Para hacer estas decisiones, habrá que exponer unas razones de por qué se ha hecho así, y las alternativas que había.

La primera de estas decisiones es la elección del sistema operativo en el que se implantará la aplicación. El sistema operativo sobre el que se implantará será Windows, por varias razones:

La primera, que todos los ordenadores de la Universidad tienen Windows instalado, pero no todos tienen otros sistemas, como Linux. Otra razón por la que he hecho esta decisión es porque esta aplicación es “puntual”, por así decirlo, ya que el número de veces que se ejecutará al año será un número bajísimo (en el mejor de los casos una vez) porque una vez se hayan generado los calendarios de exámenes del año actual no hará falta volver a ejecutarla. Por tanto, no tiene mucho sentido que también se pueda ejecutar en sistemas Linux, por ejemplo.

La tercera y última razón, y la más decisiva, es que se ha elegido el framework de .NET (como se verá más adelante) para desarrollar la aplicación, y donde mejor funciona este framework es en Windows, como es lógico.

En cuanto al lenguaje de programación para desarrollar la aplicación, ya que tenemos una aplicación compuesta de una interfaz (presentación), unas acciones que el usuario puede realizar (lógica de negocio) y un sistema de almacenamiento (nuestra base de datos), lo más eficiente es que sea un lenguaje orientado a objetos.

Por tanto, se han propuesto dos alternativas para el lenguaje a utilizar: El lenguaje de programación Java, y el framework de .NET.

Haremos un análisis de las dos alternativas, para quedarnos con la que mejor nos convenga:

- **Java:** Java es un lenguaje de programación orientado a objetos desarrollado por la empresa Sun Microsystems a principios de los años 90, aunque ahora forma parte de la empresa Oracle. Las ventajas e inconvenientes que nos ofrece este lenguaje son las siguientes:

Ventajas:

- **Portable:** El código escrito en Java es independiente de la arquitectura donde se ejecute, es por esto que se puede ejecutar en distintos sistemas operativos (“write once, run everywhere”), pero debe estar instalada en el sistema donde se vaya a ejecutar la Máquina Virtual Java (JVM), que se encarga de traducir ese código a un código nativo, para que sea capaz de ejecutarse.
- **Simple:** La sintaxis de Java es bastante simple. Además no hay que preocuparse de varias cosas que en otros lenguajes pueden ser un quebradero de cabeza como el liberar memoria, de lo que se encarga el recolector de basura, o la aritmética de punteros que no existe, así como tampoco existen macros, referencias, registros...
- **Orientado a objetos:** Soporta las tres características propias de los lenguajes orientados a objetos: encapsulación, herencia y polimorfismo.
- **Reutilizable:** Nos permite desarrollar sistemas nuevos utilizando partes desarrolladas anteriormente.

- **Seguro:** El código pasa varios test antes de ejecutarse. Se comprueba el formato de los fragmentos de código y puede detectar fragmentos de código ilegal.
- **Libre:** El lenguaje es de dominio público, y es gratis, aunque hay algunos aspectos que requieren de una licencia específica de Oracle, pero lo importante es que se pueden desarrollar aplicaciones escritas en Java de forma libre y gratuita.

Desventajas:

- **Velocidad:** Posiblemente la mayor desventaja. Es considerablemente lento frente a otros lenguajes como C++ o C#. Como el código es interpretado por la JVM la ejecución es mucho más lenta.
 - **Orientación a objetos:** Java no es un lenguaje absolutamente orientado a objetos. Por motivos de eficiencia, Java ha relajado el paradigma de orientación a objetos, y no todos los valores son objetos.
 - **Herencia múltiple:** No permite herencia múltiple como otros lenguajes.
- .NET: .NET Framework es un framework desarrollado por Microsoft con el objetivo de crear una nueva plataforma para el desarrollo de software con énfasis en la transparencia en redes, con independencia de la plataforma de hardware, y que permite un rápido desarrollo de aplicaciones. Basado en ella, la empresa intenta desarrollar una estrategia horizontal que integre todos sus productos, desde el sistema operativo hasta las herramientas de mercado. Las ventajas e inconvenientes que nos ofrece este framework son las siguientes:

Ventajas:

- **Desarrollo simplificado:** Tiene un alto nivel de abstracción, está orientado a objetos desde el núcleo, tiene un sistema de tipos unificado (CTS), y además tiene una API (Application Program Interface) organizada en forma jerárquica.
- **Multilenguaje:** Soporte múltiples lenguajes de programación: C#, C++, VB.NET, J#, Perl... Cada uno con las mismas capacidades de acceso a la plataforma y todas las facilidades.

- **Código administrado:** El CLR (Common Language Runtime) controla automáticamente el código, controla los recursos del sistema para que la ejecución se ejecute correctamente.
- **Compilación Just-In-Time (JIT):** El compilador JIT que incluye el framework compila el código intermedio generando el código máquina propio de la plataforma. Así se aumenta mucho el rendimiento de la aplicación al ser específico de cada plataforma.

Desventajas:

- **Recursos:** El consumo de recursos es mayor que Java, puede ser muy alto durante la ejecución, por eso también requiere de máquinas más potentes.
- **Multiplataforma:** No es multiplataforma, ya que solo está disponible para Windows.
- **Coste:** No cuenta con licenciamiento libre, por tanto, el desarrollar en esta plataforma genera un gran coste a las empresas, pero si se utiliza bien se puede amortizar ese coste.
- **Portabilidad:** La portabilidad es reducida, ya que solo se puede ejecutar en sistemas Windows. Aunque se han hecho implementaciones de .NET para funcionar en otras plataformas (Mono), pero la eficiencia no es la misma que la desarrollada por Microsoft.

Al final opté por desarrollarlo en .NET, en el lenguaje orientado a objetos C# por las siguientes razones:

- Se integra muy bien con Windows y todas sus herramientas, además de disponer de librerías nativas para ellas, como las proporcionadas para interactuar con Excel, lo cual es un punto favorable a tener en cuenta.
- Mi conocimiento en programación con .NET supera con creces a Java, lo que también es un factor a tener muy en cuenta.
- .NET tiene varios componentes que son muy fáciles de implementar, como la gestión de hilos de manera automática.

- La simplicidad y la alta abstracción de .NET hace que sea una gran plataforma para desarrollar, además su IDE Visual Studio es muy robusto y posee de una gran cantidad de opciones.
- La Universidad tiene licencias para poder desarrollar aplicaciones con esta tecnología, por tanto, ¿por qué no aprovecharlas?

En la elección del Sistema Gestor de Base de Datos se ha barajado entre la posibilidad de elegir Oracle, o elegir SQL Server Express. Al final se ha optado por la segunda opción porque viene incluido con el IDE Visual Studio, y porque se integra perfectamente con la plataforma elegida para desarrollar la aplicación, disponiendo de muchas facilidades para acceder y manipular la base de datos. En el apartado 6.1 se explicarán mejor estas tecnologías.

3.5 Recursos para la construcción y funcionamiento del sistema

Ahora que se han elegido las tecnologías para desarrollar nuestra aplicación, vamos a especificar qué recursos se han utilizado para desarrollar este sistema, y qué recursos se necesitarían para poder ejecutar la aplicación.

Recursos para la construcción del sistema:

- Ordenador con procesador Intel Core i5 M430 2.27 Ghz.
- Sistema operativo Windows 7.
- Microsoft Word 2007.
- Microsoft Excel 2007.
- Visual Studio 2010.
- .NET Framework 4.
- SQL Server 2008 Express Edition.
- SQL Server 2008 Management Studio.
- Microsoft Project 2007.
- Microsoft Visio 2007.

Recursos para la utilización del sistema:

- Ordenador con un procesador de, al menos, unas características similares a Intel Pentium 3 o similar.
- 512 MB memoria RAM, y 2GB libres de disco duro.
- Sistema operativo Windows XP /Vista /7.
- Tener instalado Microsoft Office Excel 2007.
- Tener instalado el .NET Framework 4.0.
- Tener instalada una instancia de SQL Server Express con la base de datos creada.

4. Gestión del proyecto

En este apartado se hablará sobre los temas relacionados con la gestión del proyecto, cómo se ha llevado a cabo, y, más concretamente, se hablará sobre el ciclo de vida del proyecto y el por qué de su elección, cómo se ha definido la planificación del proyecto en tiempo y en tareas, y qué costes ha tenido el realizar este proyecto.

Para entender todos estos aspectos, es bueno empezar definiendo lo que es un proyecto: Básicamente un proyecto se puede entender como una nueva iniciativa que consta de un principio y un final, y que está realizado por personas para alcanzar un objetivo claro y viable. Por eso el proyecto tiene que estar dentro de unos parámetros acordados de tiempo, coste y calidad.

Esto nos sirve para distinguir qué puede ser un proyecto y qué no puede serlo. En nuestro caso, tenemos una solución clara que queremos dar a un problema; es una nueva iniciativa ya que hasta ahora no estaba funcionando nada parecido; además está desarrollado por personas, que soy yo en este caso, y mi tutor con sus aportaciones; consta de un principio y un final establecidos y además se persigue conseguir un objetivo establecido, que es aportar una solución al problema propuesto mediante una aplicación software.

Ahora que ya tenemos claro en qué consiste un proyecto, y que estamos enfrente de uno, vamos a ver cómo gestionarlo, para cumplir con los objetivos propuestos del cliente y ofrecer una solución con una calidad lo suficientemente buena como para presentarla.

El tema de la gestión de proyectos es un tema muy amplio, y está enfocado principalmente a las empresas, a los recursos que utilizan, los riesgos que asumen, los costos, etc. Por eso aquí solo veremos los aspectos que más nos interesan, que es el ciclo de vida del proyecto, su planificación y los presupuestos (aunque sean ficticios).

Antes de ponernos a desarrollar el proyecto tendremos que identificar en todo lo posible, una serie de tareas necesarias de las que está formado el proyecto para dividirlo y poder gestionarlo y desarrollarlo mejor. Esto nos sirve para modularizar el proyecto y tener más claros los objetivos parciales, para poder llegar al objetivo final.

4.1 Ciclo de vida

En este apartado definiremos en qué consiste el ciclo de vida del proyecto, cómo definiremos uno que sea adecuado, y las características de este ciclo de vida.

Como hemos comentado en el apartado anterior, un proyecto es un proceso temporal y como tal, está fragmentado en una serie de etapas por las que hay que ir pasando. Cada una de estas etapas tiene un objetivo propio: se completa con la revisión de un subproducto, y da paso a la siguiente etapa.

El ciclo de vida se define como el conjunto de estas etapas en que se divide el proyecto. De forma general se pueden definir estas etapas como sigue: Iniciación, planificación / programación, ejecución, control y cierre.

En la etapa de iniciación se define la necesidad real del proyecto, además del producto final. También se determinan las distintas prioridades del proyecto, como son los objetivos parciales a cumplir, y la documentación inicial del proyecto.

En la etapa de planificación / organización se ordenan las actividades de forma lógica y se preparan los elementos necesarios para hacer posible el proyecto. Esta etapa es una de las más importantes porque hay que definir varios aspectos, como son la planificación de los objetivos de acuerdo a los recursos disponibles, las tareas y su duración, las personas involucradas y su carga, los plazos, los costes, y los riesgos a los que tiene que hacer frente la organización. Nosotros veremos más adelante los aspectos que nos son más relevantes y que son los recursos, las tareas, los plazos y los costes.

La fase de ejecución y control es la fase intermedia donde se implementa el sistema y se comparan los datos reales con los datos estimados, además se analiza el impacto que esta diferencia puede tener y, si se da el caso, se procederá a una acción correctiva y se revisará la planificación del proyecto.

En la última fase, la de cierre, se recogen los datos finales y se entrega el producto final al cliente. Además se hace una revisión final del proyecto y un cierre documental del mismo. De este modo, se podrán conocer los errores cometidos y como mejorarlos para la siguiente vez.

Estas fases del ciclo de vida del proyecto son genéricas, pero son necesarias para conocer conceptualmente en qué consiste cada fase. En la actualidad, lo que se entiende por ciclos de vida del software son implementaciones más específicas de este modelo, aplicado al software, y hay de varios tipos. Algunos de estos tipos son más utilizados por las organizaciones y otros menos, también dependiendo del tipo de proyecto.

Pero hay dos modelos básicos: Los modelos basados en el ciclo de vida en cascada o “modelos clásicos”, y los modelos evolutivos. Las organizaciones aún utilizan ampliamente el modelo en cascada, en realidad cada uno tiene sus ventajas y sus inconvenientes. A continuación explicaremos brevemente estos dos grandes modelos y sus características, para luego justificar la elección del ciclo de vida que hemos hecho.

Modelos en cascada:

El modelo en cascada puro raramente se utiliza ya, ya que no hay una retroalimentación en las etapas, por lo que requiere de un conocimiento absoluto sobre los requisitos y la rigidez de los mismos (lo cual es prácticamente imposible porque el software es evolutivo), por tanto algún cambio durante la ejecución de alguna de las etapas implicaría volver al principio, dando como resultado grandes costes de tiempo y desarrollo. Este modelo puro se podría aplicar a pequeños sistemas.

Pero hay muchas variantes de este modelo para hacerlo más dinámico y ajustarse a los sistemas modernos, y el más utilizado es el modelo en cascada con retroalimentación. En este modelo cada etapa tiene retroalimentación, es decir, se puede ir hacia atrás y hacia delante en las etapas. Esto da lugar a la oportunidad de desarrollar software en el que hay cambios en los requisitos, matices o evoluciones en el sistema, que es lo que ocurre normalmente. Este modelo es uno de los más usados y populares.

Modelos evolutivos:

Este modelo cuenta con que los requisitos suelen cambiar conforme se desarrolla el software, lo que suele ser habitual. Está diseñado para acomodarse a una evolución temporal. Los requisitos centrales están bien definidos, pero no conocidos al detalle. Estos modelos permiten desarrollar versiones cada vez más completas hasta llegar a la solución final, porque los plazos del mercado y de la competencia hacen que no sea posible sacar el producto completo en un principio, siendo aconsejable introducir una versión funcional y limitada, antes de introducir la solución final.

También hay varias variantes de este modelo, pero se explicará la más relevante, que es el ciclo de vida en espiral. Este modelo consiste en una serie de ciclos que se van repitiendo. Cada uno de estos ciclos tiene las mismas fases y cuando se termina un ciclo, da lugar a un producto ampliado con respecto del ciclo anterior.

Es parecido al modelo incremental básico, pero se añade además un factor importante, el concepto de riesgo. Un riesgo puede ser, por ejemplo, unos requisitos mal comprendidos, un diseño ineficiente, errores de implementación, etc.

Al terminar una iteración se comprueba que el producto cumple con los requisitos establecidos para ese ciclo, y el propio cliente evalúa el producto. No existe una diferenciación muy clara entre cuando termina el proyecto y cuando empieza la fase de mantenimiento, cuando hay que hacer un cambio, este puede consistir en una nueva iteración.

Elección:

Para abordar este proyecto se ha elegido el ciclo en cascada con retroalimentación, principalmente por las dos razones siguientes:

- Es más fácil de implementar, la planificación es mucho más sencilla.
- Desde un principio teníamos muy claros los objetivos y requisitos que había que cumplir, por lo que este método es una buena opción.

A continuación se muestra un esquema del ciclo de vida elegido:

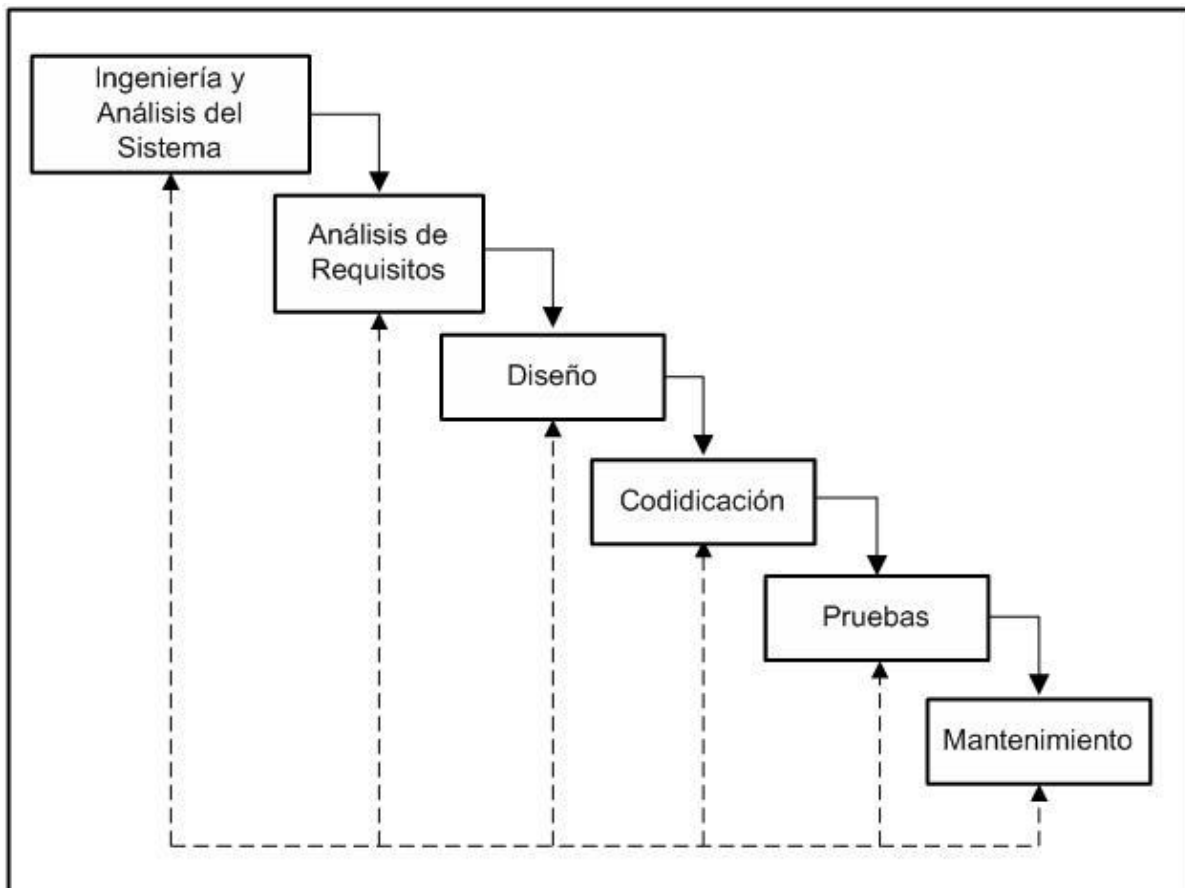


Ilustración 2: Ciclo de vida en cascada con retroalimentación

4.2 Planificación y presupuestos

Dentro de este apartado englobaremos todo lo referente a la fase de planificación / organización explicada anteriormente.

Como hemos dicho esta etapa es una de las más importantes, y aquí trataremos el tema de los recursos, las tareas y los plazos que hay que establecer para desarrollarlas.

En cuanto a los recursos podemos distinguir entre recursos materiales y humanos. En este caso se podría distinguir un recurso material, que es el ordenador de trabajo, y un solo recurso humano, que en este caso sería mi propia persona. Los recursos se asignan a las tareas previamente creadas. Una vez que se asignan los recursos, se comprueba mediante histogramas de recursos que ningún recurso está sobreasignado. En este caso particular eso no tendría sentido, ya que solo tenemos un recurso en el proyecto.

El proyecto se dividirá en varias tareas donde cada tarea tiene una duración asignada en días, y una fecha de comienzo y una fecha de fin. Estas tareas se organizarán en el tiempo e irán en un orden específico a fin de ir pasando por las diferentes fases de un modo correcto.

Se especificarán estos valores teniendo en cuenta lo siguiente: Todas las tareas han sido llevadas a cabo por una sola persona, un día de trabajo equivale a cuatro horas de trabajo, y los días de trabajo han sido de lunes a sábado. Se ha tomado el supuesto de que esa persona cobra 7.5 euros por hora.

Por tanto, el esfuerzo de una semana de trabajo es:

$$(1 \text{ persona} \times 6 \text{ días} \times 4 \text{ horas}) / 1 \text{ semana} = 24 \text{ horas} / \text{semana}$$

En la siguiente página se mostrará la el desglose de tareas y sus duraciones y fechas en un cuadro para posteriormente mostrarlas en un Diagrama de Gantt.

A continuación se muestra el desglose de tareas de las que consta el proyecto y su duración, así como de las tareas que hay que cumplir antes que otras (predecesoras). También se puede observar los recursos de los que consta cada tarea, que en este caso, es solo uno y siempre el mismo, el programador.

Id	Nombre de tarea	Duración	Comienzo	Fin	Predecesoras	Nombres de los recursos
1	Inicio del proyecto	0 días	lun 02/01/12	lun 02/01/12		Programador - Rubén
2	Especificación de requisitos	30 días	lun 02/01/12	sáb 04/02/12		Programador - Rubén
3	Establecimiento de objetivos generales	7 días	lun 02/01/12	lun 09/01/12		Programador - Rubén
4	Requisitos de usuario	15 días	lun 02/01/12	mié 18/01/12		Programador - Rubén
5	Casos de uso	10 días	jue 19/01/12	lun 30/01/12	4	Programador - Rubén
6	Requisitos de software	12 días	lun 16/01/12	sáb 28/01/12		Programador - Rubén
7	Estudio de viabilidad	5 días	mar 31/01/12	sáb 04/02/12	5;6	Programador - Rubén
8	Análisis y Diseño	21 días	lun 06/02/12	mié 29/02/12	2	Programador - Rubén
9	Organización de la información	12 días	lun 08/02/12	sáb 18/02/12		Programador - Rubén
10	Análisis funcional	6 días	lun 08/02/12	sáb 11/02/12		Programador - Rubén
11	Diseño de las interfaces	15 días	lun 13/02/12	mié 29/02/12	10	Programador - Rubén
12	Implementación	105 días	jue 01/03/12	sáb 30/06/12	8	Programador - Rubén
13	Implementación de la Base de Datos	7 días	jue 01/03/12	jue 08/03/12		Programador - Rubén
14	Creación del modelo físico	6 días	jue 01/03/12	mié 07/03/12		Programador - Rubén
15	Implementación del modelo	1 día	jue 08/03/12	jue 08/03/12	14	Programador - Rubén
16	Definición de los componentes	38 días	vie 09/03/12	sáb 21/04/12	13	Programador - Rubén
17	Definición del componente Dominio	24 días	vie 09/03/12	jue 05/04/12		Programador - Rubén
18	Definición del componente LogicaNegocio	38 días	vie 09/03/12	sáb 21/04/12		Programador - Rubén
19	Definición del componente Presentación (GUI)	20 días	vie 09/03/12	sáb 31/03/12		Programador - Rubén
20	Implementación de los componentes	98 días	vie 09/03/12	sáb 30/06/12	13	Programador - Rubén
21	Implementación del componente Dominio	52 días	vie 09/03/12	mar 08/05/12		Programador - Rubén
22	Implementación del componente LogicaNegocio	98 días	vie 09/03/12	sáb 30/06/12		Programador - Rubén
23	Implementación del componente Presentación (GUI)	65 días	vie 09/03/12	mié 23/05/12		Programador - Rubén
24	Pruebas de aceptación	3 días	lun 02/07/12	mié 04/07/12	12	Programador - Rubén
25	Creación del manual de usuario	1 día	jue 05/07/12	jue 05/07/12	24	Programador - Rubén
26	Creación de la presentación	4 días	vie 08/07/12	mar 10/07/12	25	Programador - Rubén
27	Fin del proyecto	0 días	mar 10/07/12	mar 10/07/12	26	Programador - Rubén

Ilustración 3: Listado de tareas

A continuación se mostrará el diagrama de Gantt, que es una representación de todas las tareas en el tiempo. En esencia es un diagrama de barras, que visualiza gráficamente las relaciones en tiempo de las diferentes tareas del proyecto:

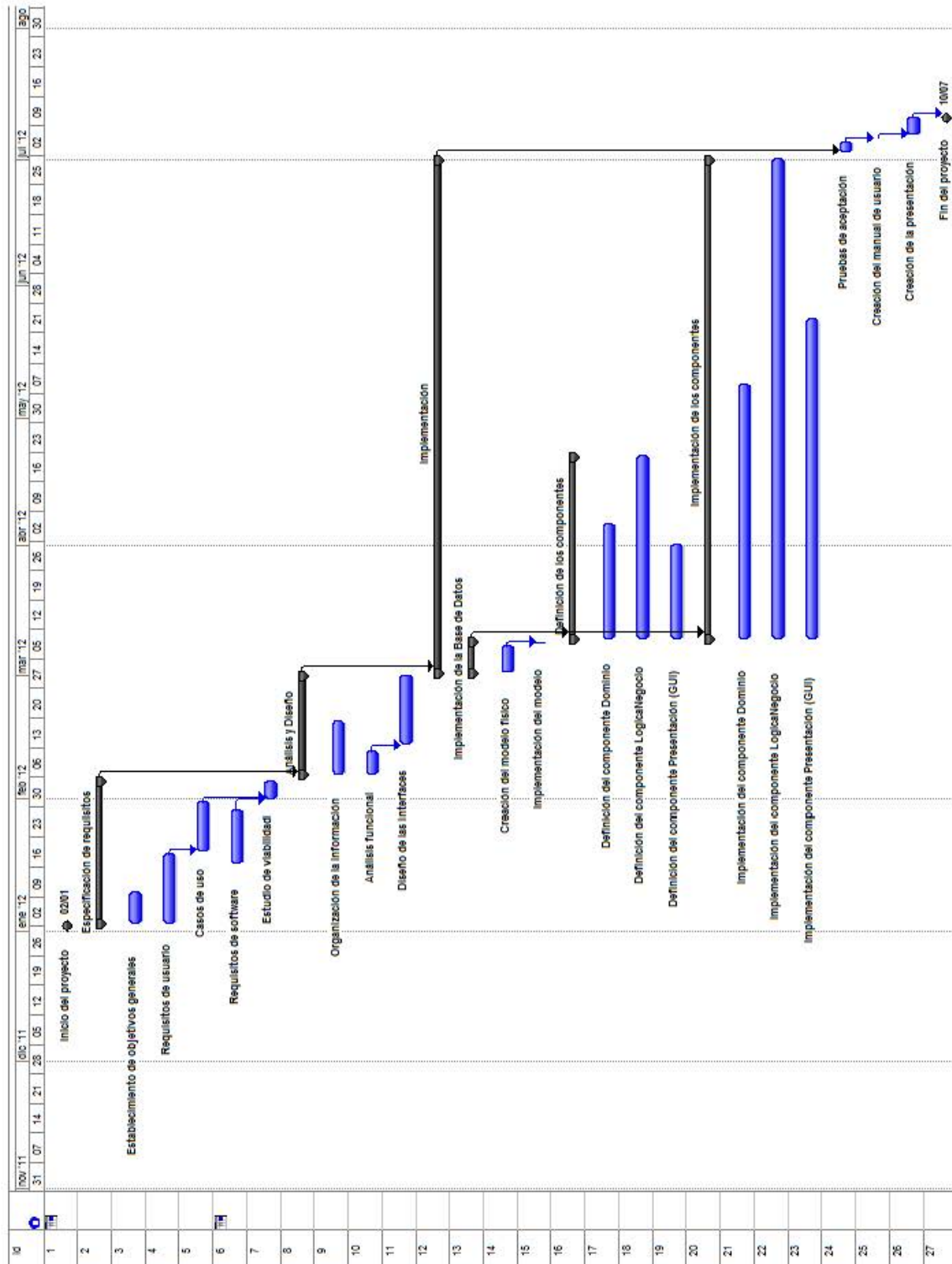


Ilustración 4: Diagrama de Gantt

En esta hoja se mostrarán los datos relativos al presupuesto del proyecto, desglosado en costos por tareas. Para ello se utilizará el esfuerzo que se ha comentado en la planificación anterior.

Id	Nombre de tarea	Costo total
1	Inicio del proyecto	0 €
2	Especificación de requisitos	900 €
8	Análisis y Diseño	157.5 €
12	Implementación	3150 €
24	Pruebas de aceptación	90 €
25	Creación del manual de usuario	30 €
26	Creación de la presentación	120 €
27	Fin del proyecto	0 €

Ilustración 5: Asignación de costes

Vemos que donde hay mayores costes es en la fase de implementación, esto es lógico, porque es la fase donde se invierten más horas en el proyecto.

Por tanto, en esta simulación el presupuesto del proyecto ascendería a 4447.5 euros. La duración total en días del proyecto ha sido de 190 días (alrededor de 6 meses y 10 días).

5. Análisis y diseño de la aplicación

En el presente apartado se definirá el análisis de ciertos requisitos de software que implican el realizar un diseño adecuado, tanto del almacenamiento de datos como de la funcionalidad de la aplicación.

Se dividirá en: las necesidades de la información, donde se tratará el tema del diseño de la Base de Datos y el por qué de ese diseño; el análisis funcional, donde se tratará el tema del diseño de la aplicación software y la funcionalidad y objetivos de cada componente; y por último, se tratará el diseño de las interfaces que se ha elegido, además de justificarlo.

El diseño arquitectural solo puede comenzar una vez que se tenga un entendimiento más o menos claro del sistema a implementar, aunque el diseño, como el análisis, nunca termina realmente hasta que el producto final es entregado.

El diseño se enfoca en la estructura, dinámica y estática, y su principal propósito es el de crear el esqueleto del sistema, sobre el cual se basa toda la implementación. Esta fase se suele definir como la fase más importante de un desarrollo.

Si realizamos un mal diseño se irán arrastrando los problemas que esto genera a todas las fases del desarrollo, obteniendo como resultado un software de mala calidad o, al menos, de no tan buena calidad.

A continuación haremos un análisis de los datos de la aplicación. Se realizará un estudio de toda la información que tiene que ser manejada por la aplicación, detallando aquella información que debe ser almacenada de forma persistente para cumplir con los objetivos establecidos.

5.1 Necesidades de la información

Para poder saber qué información debe manejar la aplicación y la forma de almacenamiento debemos realizar un análisis de algunos de los requisitos que se han establecido.

Como ya se ha comentado anteriormente, el listado formal de la totalidad de los requisitos se puede encontrar en el “Anexo B: Listado de requisitos”.

A continuación se exponen aquellos requisitos que hacen referencia al manejo y almacenamiento de datos:

RU-10:

La aplicación deberá disponer de una entrada de datos para el año actual, donde además de estar los datos relativos a las fechas de los exámenes de ADE (que nos vienen impuestos), se encontrarán los datos relativos a las asignaturas de las titulaciones. Esta entrada deberá tener un formato establecido.

RU-23:

La aplicación deberá manejar la información relativa a las asignaturas. Además, deberá guardarse toda esa información en almacenamiento persistente, para hacer la aplicación más funcional. Estos datos referentes a las asignaturas son los siguientes:

- Código de la asignatura.
- Nombre de la asignatura.
- Titulación de la asignatura.
- Curso de la asignatura.
- Cuatrimestre de la asignatura.
- Especialidad de la asignatura.
- Si es optativa o no.
- Fecha de examen de la asignatura.
- Fecha de examen extraordinario de la asignatura.
- Dificultad de la asignatura.

RU-01, RU-02:

Los datos relativos a las fechas de inicio y de fin de los distintos períodos de exámenes no deberán de almacenarse porque el usuario podrá elegirlos manualmente en la aplicación.

RU-23:

Se tendrá que poder almacenar el año al que hacen referencia las asignaturas de entrada de la aplicación. En concreto deberán guardarse los siguientes datos relativos al año:

- Descripción del año, que consistirá en el curso actual, es decir, el año actual y el siguiente año.

RU-04:

Se tendrá que poder almacenar los distintos períodos a los que pueden pertenecer las asignaturas de la aplicación. En concreto deberán guardarse los siguientes datos relativos al período:

- Nombre del período: Solo habrá dos períodos a los que puedan pertenecer las asignaturas, pueden pertenecer al primer cuatrimestre o al segundo cuatrimestre.

RU-03:

Se tendrá que poder almacenar los distintos cursos a los que pueden pertenecer las asignaturas de la aplicación. En concreto deberán guardarse los siguientes datos relativos al curso:

- Número de curso: Los distintos cursos son los siguientes: Primero (1), Segundo (2), Tercero (3), Cuarto (4), Quinto (5), Sexto (6). Esto es así porque el Doble Grado en Informática – ADE tiene seis cursos, puesto que es un doble grado.

RU-05:

Se tendrá que poder almacenar las titulaciones a los que pueden pertenecer las asignaturas de la aplicación. En concreto deberán guardarse los siguientes datos relativos a la titulación:

- Nombre de la titulación.


Ahora que tenemos los distintos requisitos que hacen referencia al almacenamiento y manejo de información que utiliza la aplicación, procederemos a diseñar el modelo de almacenamiento de datos en la siguiente página.

Definición de tablas

El diseño de la Base de Datos se ha definido de la manera que nos permita cumplir los requisitos vistos anteriormente, y también de la manera más abierta posible, es decir, intentando mantener un diseño estándar por si en un futuro se implementa la aplicación con la posibilidad de que haya más titulaciones, o más cursos, u otro tipo de información.

A continuación se muestran las distintas entidades para cumplir estos requisitos de manera satisfactoria:




- Tabla Titulación:

Titulacion	
	Id
	Nombre


- Tabla Cuatrimestre:

Cuatrimestre	
	Id
	Nombre

- Tabla Asignatura:

Asignatura	
	Codigo
	IdTitulacion
	IdCurso
	IdCuatrimestre
	Nombre
	Especialidad
	Subtitulacion
	Optativa
	FechaExamen
	FechaExtraordinaria
	Peso

- Tabla Año:

Año	
 Id	
Descripcion	

- Tabla Curso:

Curso	
 Id	
Nombre	

Estas tablas son las que forman la totalidad de la Base de Datos “CalendarioExámenes”, y son las que guardaran los valores que deben estar almacenados para el correcto funcionamiento de la aplicación.

Definición del modelo E/R

Ahora que hemos definido cada tabla por separado, vamos a proceder a crear el modelo Entidad Relación que nos muestra las distintas relaciones entre las tablas. Esto nos ayudará para entender mejor los requisitos explicados anteriormente.

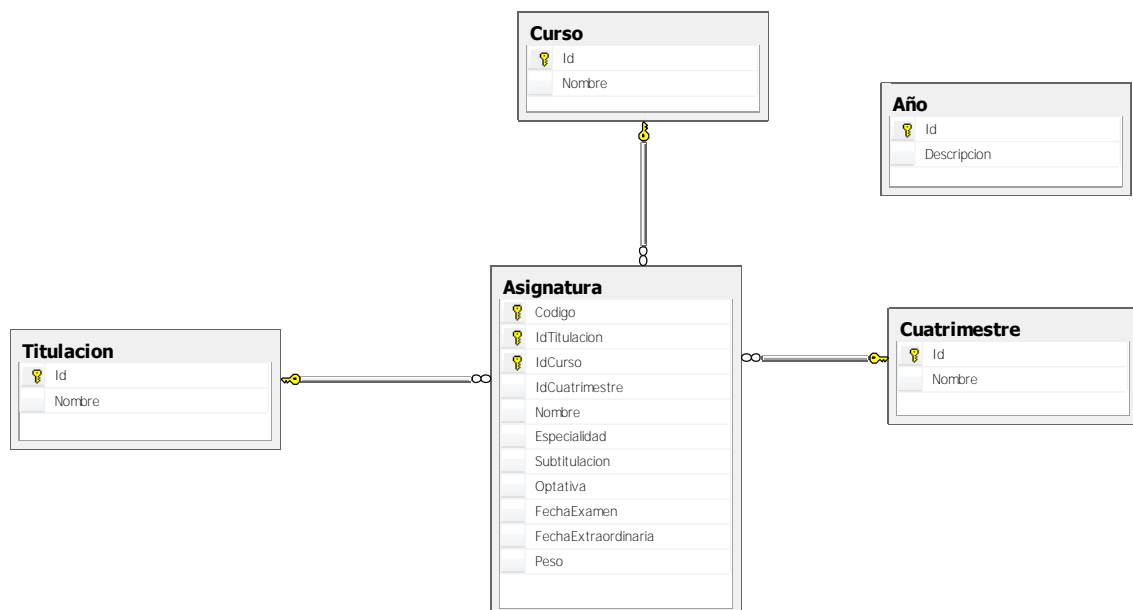


Ilustración 6: Modelo E/R

Explicación del modelo:

En este esquema las “llaves” hacen referencia a la multiplicidad 1, y los “infinitos” a la multiplicidad múltiple. Esto significa que una titulación podrá tener muchas asignaturas, pero una asignatura sólo pertenece a una titulación. Podemos pensar que esto no es así por las asignaturas de los dobles grados, que pertenecen a dos titulaciones, pero en realidad un doble grado se tiene en cuenta como una titulación aparte, y no de dos separadas, por lo que sí se cumple.

El cuatrimestre hace referencia al período que hemos dicho anteriormente (RS-05). Un cuatrimestre puede tener varias asignaturas pero una asignatura es de un solo cuatrimestre.

La entidad Curso también está relacionada con Asignatura, ya que un curso tiene varias asignaturas, pero una asignatura solo puede pertenecer a un curso. Esto en realidad no tiene por qué ser así, por eso la clave principal de la tabla Asignatura la forman tres valores: el código de asignatura, la titulación, y el curso. Esto es debido a los dobles grados de nuevo, ya que asignaturas que son de un curso en el doble grado pueden ser de un curso distinto en su titulación ordinaria.

En la entidad Asignatura tenemos que explicar dos campos: la subtitulación y el peso. El campo subtitulación se ha añadido para identificar de que titulación ordinaria es una asignatura de un doble grado, en este caso para distinguir si es una asignatura de Informática o de ADE, ya que tenemos que poder distinguirlo de alguna manera para la aplicación. El peso hace referencia a la dificultad de la asignatura de la que hemos hablado en el requisito RS-02, para saber si una asignatura es difícil o no. Más adelante hablaremos de los posibles valores de los campos con el formato del fichero de entrada.

Por último, como podemos observar, la entidad Año no tiene relación alguna con ninguna otra entidad, ya que no la necesita. Almacenará solo la información relevante al año en el que se encuentran los datos de las asignaturas.

Semántica explícita:

En el esquema E/R hay restricciones definidas dentro de las entidades que el esquema no puede reflejar. Pasaremos a definirlas:

Para la entidad Titulación:

- El campo Id es numérico y obligatorio (en realidad esto se puede suponer ya que es clave primaria de la entidad).

Para la entidad Curso:

- El campo Id es numérico y obligatorio (en realidad esto se puede suponer ya que es clave primaria de la entidad).

Para la entidad Año:

- El campo Id es numérico y obligatorio (en realidad esto se puede suponer ya que es clave primaria de la entidad).

Para la entidad Asignatura:

- El Id de esta tabla estará formado por tres campos: el código de la asignatura, la titulación y el curso.
- El código de la asignatura es un valor numérico es un valor de cinco cifras, con dos excepciones que se verán más adelante.
- El campo optativa será de tipo “bit”, es decir, solo admitirá valores “true” o “false”.
- El campo FechaExamen será de tipo “datetime”, es decir fecha y hora, ya que la fecha definirá el día del examen y la hora si es del turno de tarde o de mañana.
- El campo FechaExtraordinaria será de tipo “datetime”, es decir fecha y hora, ya que la fecha definirá el día del examen y la hora si es del turno de tarde o de mañana.
- El campo Peso será de tipo numérico.

Es necesario definir estas restricciones para saber de qué tipos de valores están formados los distintos campos de las entidades. Las dos excepciones en cuanto al código de la asignatura se deben a dos asignaturas que tienen un “tratamiento especial”. Estas asignaturas son “Matemática discreta” y “Álgebra lineal”. Lo que ocurre con estas asignaturas es que son compartidas por las dos titulaciones pero sin embargo no comparten cuatrimestre, es el único caso que existe que cumple esto, por eso son especiales, y tienen que tener un código modificado, que es su código de cinco dígitos más otro, que será un 1.

Diseño del fichero de entrada

A continuación pasaremos a explicar el diseño del fichero de entrada donde vienen los datos de las asignaturas para cargar a la Base de Datos. Este fichero será un Excel por la razón de que es fácil modificar y se le puede aplicar unos filtros para que el usuario solo pueda insertar unos valores específicos en los datos de las asignaturas.

Esto tiene como finalidad que el usuario no pueda insertar valores no válidos en algún campo de las tablas de la Base de Datos.

Además se ha elegido un fichero Excel por la razón de que esos datos de asignaturas vendrán ordenados por curso, y así es fácil hacer un seguimiento, e incluso insertar asignaturas nuevas por si el día de mañana cambia el plan de estudios y alguna asignatura cambia (también se pueden eliminar).

También es la manera más legible que tiene el usuario para ver los datos que va a cargar a la Base de Datos y poder modificar sus valores también. Este fichero tendrá una hoja por titulación, ya que cambiará el formato. A continuación, se muestran los distintos formatos del fichero para cada titulación:

CURSO	CUATRIMESTRE	CÓDIGO	NOMBRE	ESPECIALIDAD	OPTATIVA	PESO
-------	--------------	--------	--------	--------------	----------	------

CURSO	CUATRIMESTRE	CÓDIGO	NOMBRE	ESPECIALIDAD	SUBSTITUCION	PESO	FECHA EXAMEN	TURNO	FECHA EXTRAORDINARIA	TURNO
-------	--------------	--------	--------	--------------	--------------	------	--------------	-------	----------------------	-------

La columna “Turno” aparece porque es necesario decir si el examen de la asignatura será en el turno de mañana o en el turno de tarde (M o T). En la Base de Datos esto se implementa en un solo campo como hemos visto antes.

También habrá una tercera hoja donde irán los filtros para los valores que hemos comentado antes especificando que el usuario no deberá modificar, ya que se supone que estos filtros no van a cambiar con el tiempo.

Diseño del calendario generado

En esta hoja explicaremos el diseño del fichero/ficheros de salida de la aplicación, es decir, los calendarios de exámenes. Este formato deberá ser un formato legible y amigable para el usuario de manera que lo vea como un calendario por semanas. Cada día tendrá un espacio para el turno de mañana y otro para el turno de tarde.

Además, en este calendario de salida deberá ir el curso académico actual del calendario de exámenes, el período generado, y para que cursos se ha generado.

Para cada asignatura se mostrarán los siguientes datos en este calendario:

- Código de la asignatura.
- Curso de la asignatura.
- Nombre de la asignatura.
- Especialidad de la asignatura.
- Subtitulación (Opcional), solo si la asignatura es del doble grado.

De esta manera, obtendremos un calendario simple pero a la vez con toda la información que nos es necesaria, además de bien presentado.

En cada calendario solo podrá ir un curso, o todos a la vez, de una titulación, tal y como se especificará en el listado formal de los requisitos más adelante.

5.2 Análisis Funcional

En este apartado se realizará un análisis de aquellos requisitos del software que nos ayuden a saber cómo hemos de implementar los distintos componentes software de los que constará nuestra aplicación, de manera que satisfagamos esas necesidades que nos vienen impuestas en esos requisitos.

Esto nos ayudara a saber cuál es la arquitectura más eficiente para dar solución a estas necesidades y como organizar esos componentes de la manera más adecuada.

Siempre podrán rectificarse todos los valores que introduce el usuario en cualquier momento durante la ejecución de la aplicación, por si se ha equivocado anteriormente sin darse cuenta, y lo ha recordado más adelante.

Así pues, procederemos a describir aquellos requisitos que necesitamos para este propósito:

RS-01:

Necesidad de una interfaz gráfica donde el usuario pueda interactuar con todas las funcionalidades de la aplicación.

RS-08:

Necesidad de poder elegir de una forma gráfica las optativas que se vayan a impartir el año actual, o todas si se desea.

RS-10, RS-11:

Necesidad de poder mostrar de forma gráfica todas las asignaturas de las que constan las titulaciones.

RS-16:

El usuario podrá cancelar en cualquier momento el proceso de generación del calendario durante su ejecución si lo desea, por si se ha equivocado al establecer algún valor, como las fechas por ejemplo.

RS-12:

La interfaz deberá tener un diseño lo más claro, simple y a la vez presentable para el usuario, y también deberá tener un manejo sencillo y rápido.

Arquitectura funcional:

Una vez hemos descrito estos requisitos que nos afectarán para el diseño de la arquitectura a elegir y los elementos que habrá que desarrollar, procederemos a explicarlos.

Como ya hemos visto, necesitamos una aplicación que sea gráfica, con una interfaz sencilla y legible con la que el usuario podrá interactuar cómodamente, y de un Sistema Gestor de Base de Datos donde se almacenará la información relativa a las asignaturas de las titulaciones. También habrá que tener en cuenta el fichero de entrada de la aplicación, así como las salidas que se pueden generar (que serán calendarios).

Por todo esto se ha elegido una arquitectura de tres capas: Presentación, negocio y datos. La capa de presentación se encargará de mostrar al usuario la interfaz gráfica con la que podrá interactuar. La capa de negocio tratará la lógica de la aplicación, con las acciones que el usuario realiza, desde la capa de presentación. Por último, la capa de datos se encargará de manejar el sistema de la Base de Datos, acceder a él y modificar los datos si es preciso.

La implementación de estas tres capas se verá en el apartado seis, pero se mostrará el siguiente dibujo, que ilustra muy bien el propósito de estas tres capas:

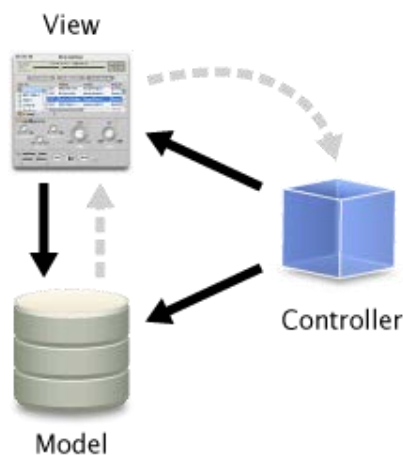


Ilustración 7: Modelo de tres capas

A continuación veremos un diagrama con nuestros componentes, que corresponderán a las tres capas que hemos explicado anteriormente. Así pues, tendremos tres componentes: Dominio (capa de datos), GUI (capa de presentación) y Lógica de Negocio (que corresponderá a la capa del controlador).

El diagrama de componentes de la aplicación es el siguiente:

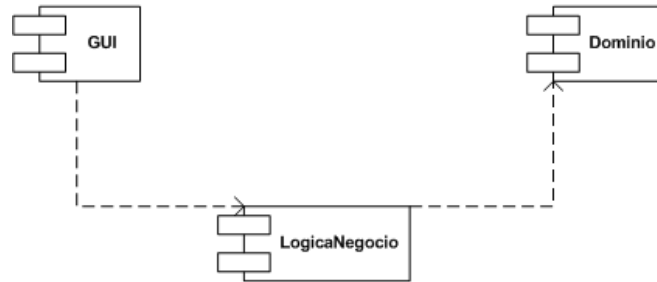


Ilustración 8: Diagrama de componentes

Una vez definido nuestro diagrama de componentes, se mostrará el diagrama de clases, que es el diseño de las distintas clases de la aplicación, porque como ya hemos comentado con anterioridad (en el apartado 3), se utilizará un lenguaje orientado a objetos por los requerimientos de la aplicación, ya que es una aplicación de alto nivel donde se pueden diferenciar bien las tres capas.

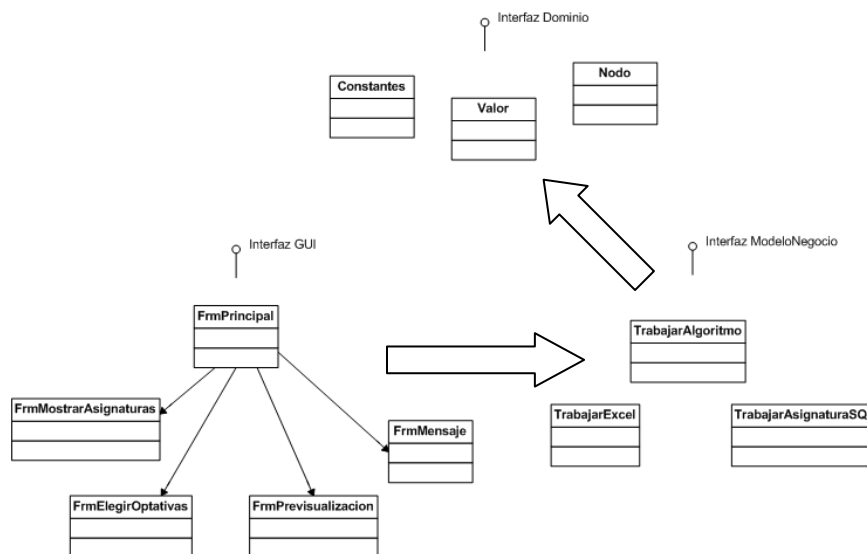


Ilustración 9: Diagrama de clases

5.3 Diseño de interfaces

Aquí definiremos todas las ventanas y los elementos relevantes de las mismas. Para esta labor, es necesario conocer los requerimientos vistos anteriormente, tanto los requisitos de usuario como los requisitos de software, para hacer un buen diseño del aspecto gráfico de la aplicación.

También es necesario conocer el usuario final de la aplicación. El usuario final será un profesor de la Universidad, por tanto estará familiarizado con aplicaciones gráficas tipo Windows, y sus respectivos componentes.

Por tanto, para cumplir con los requisitos establecidos la aplicación deberá estar formada por:

- Una ventana principal, donde se podrán realizar la mayoría de funcionalidades de la aplicación.
- Una ventana distinta para mostrar asignaturas según distintos criterios.
- Una ventana de previsualización de los resultados obtenidos.
- Una ventana que mostrará un mensaje de espera cuando se esté realizando la lectura del fichero de entrada.
- Una ventana para poder realizar la funcionalidad de elegir unas determinadas asignaturas optativas a impartir este año.

En la siguiente página veremos el flujo de navegación que se puede dar en la aplicación, más bien todos ellos. Las ventanas aparecerán en miniatura, porque lo que queremos reflejar son los distintos flujos de navegación por ventanas que tiene la aplicación para, más adelante, mostrar el diseño de los componentes relevantes.

Las ventanas en tamaño normal se verán en el “Anexo D: Manual de Usuario” porque aquí es donde se explicarán todas las funcionalidades de las ventana para el usuario final.

Flujo entre las distintas ventanas de la aplicación. Cada ventana viene identificada por un número, con la siguiente correspondencia:

- Ventana principal: Paso 1 → 1
- Ventana de mensaje de espera (leyendo fichero de entrada) → 2.
- Ventana de elección de optativas → 3.
- Ventana de mostrar optativas elegidas → 4.
- Ventana de mostrar todas las asignaturas de Informática → 4 (es el mismo número porque el diseño de la ventana es el mismo).
- Ventana de mostrar todas las asignaturas del Doble Grado en Inf – ADE → 4 (es el mismo número porque el diseño de la ventana es el mismo).
- Ventana principal: Paso 2 → 5.
- Ventana principal: Paso 3 → 6.
- Ventana de previsualización de resultado → 7.
- Ventana principal: Paso 4 → 8.

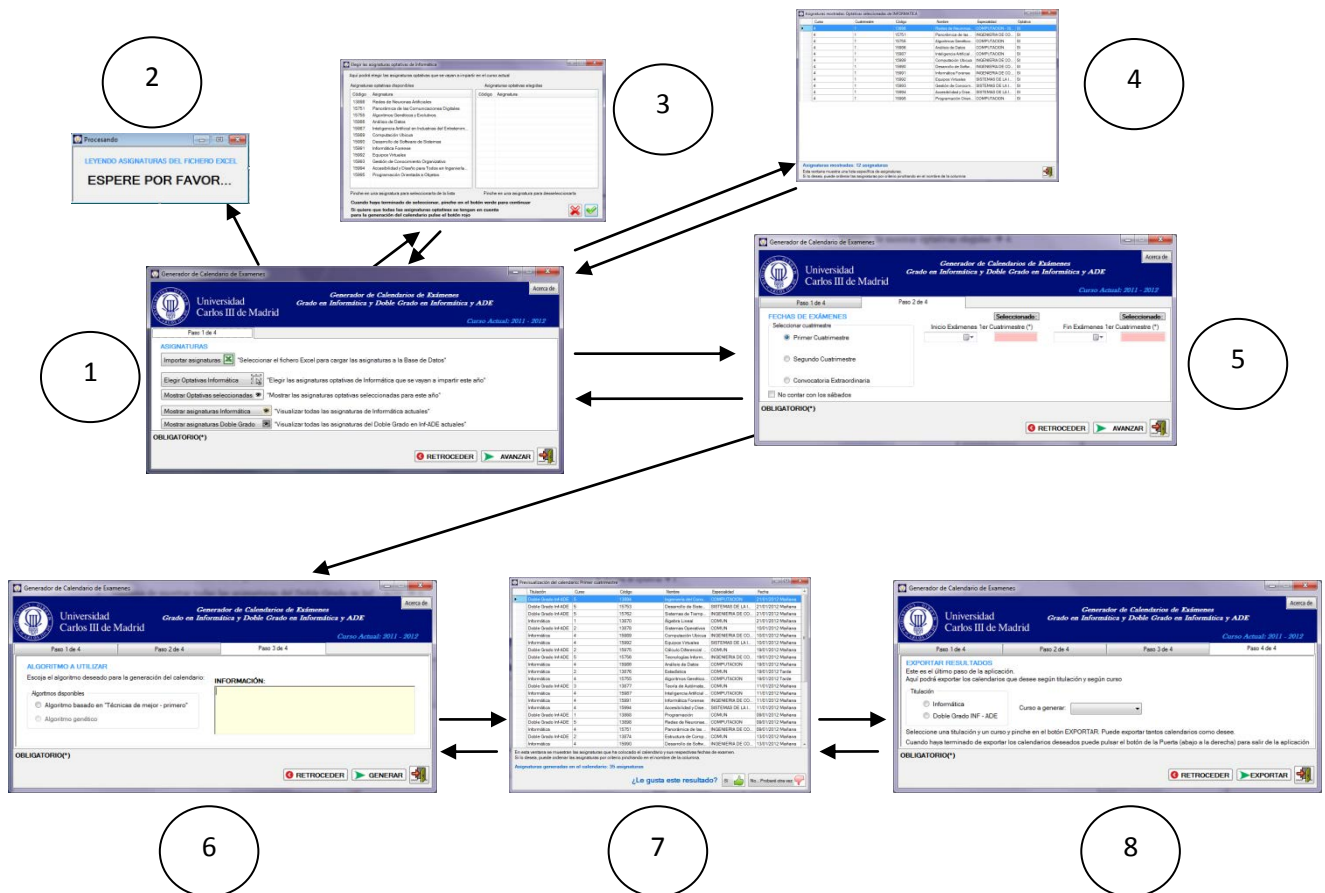


Ilustración 10: Flujo entre pantallas

Como podemos observar en la página anterior, la aplicación se ha diseñado de manera que hay una ventana principal, donde se realiza la mayor parte de la funcionalidad del sistema, ya que está dividida por pestañas. Cada una de estas pestañas será un “paso” que hay que dar hasta llegar al final, donde se pueden exportar los calendarios. Esto hace que la aplicación sea más simple y más rápida.

Además, independientemente del paso en el que estés, puedes ir hacia el paso siguiente o el paso anterior, incluso aunque estés en el final, aumentando la funcionalidad del sistema. Esto se ha diseñado así para no dar lugar a tener que volver a ejecutar la aplicación si el usuario se equivoca, o si no le gusta el calendario por ejemplo.

También hay que decir que la aplicación se ha diseñado con una interfaz seria, pero también elegante, para que al usuario le resulte agradable utilizarla, además es una interfaz bastante limpia.

En la ventana principal se muestra el logo de la Universidad entre una franja del mismo color azul, el fondo de las ventanas es de color gris claro, que es poco agresivo y agradable a la vista, por lo que cansa menos.



Ilustración 11: Franja de la ventana principal

En la barra de título de la aplicación también está el icono con el logo de la Universidad en miniatura, para que aparezca en la barra de tareas también, lo que hace que la aplicación sea más elegante.



Ilustración 12: Barra de título de la aplicación

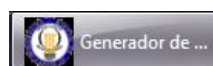


Ilustración 13: Aplicación en la barra de tareas

Se procederá a exponer diferentes elementos de las interfaces que también nos son relevantes. La intención no es poner todos los elementos de las pantallas, solo aquellos que sean más relevantes, ya que en el manual de usuario ya se mostrarán mejor todos estos elementos acompañados de la explicación de su funcionalidad para el usuario.

Si no sería bastante redundante ponerlo aquí, y nuestro objetivo no es ese.

Se explicará el funcionamiento de las pestañas de la ventana principal, es el siguiente: La navegación será mediante botones única y exclusivamente. Además las pestañas de los pasos futuros no se mostrarán al usuario, solo se mostrará la pestaña del paso actual, y las de los anteriores si los hay. Con esto se consigue dar al usuario, la sensación de “ir avanzando paso por paso” mediante los botones, mientras se va pasando de pestaña en pestaña. El diseño de las pestañas es el siguiente:



Ilustración 14: Diseño de las pestañas (pasos) de la pantalla principal

También hay que decir que todos los componentes de las ventanas están explicados con un texto descriptivo, como los botones, los RadioButton, etc. Por si esto fuera poco, en los componentes donde no exista esa descripción o dé lugar a equívocos también habrá globos de ayuda cuando se deja el ratón por encima un momento. Esto también es una manera de ayudar al usuario a saber para qué sirve cada componente.

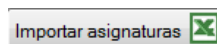


Ilustración 15: Diseño del botón para importar las asignaturas del fichero de entrada



Ilustración 16: Ejemplo de globo de ayuda para el usuario

6. Implementación de la aplicación

Una vez realizados el análisis y el diseño de la aplicación ya podemos proceder a la fase que suele ser la más larga en duración del proyecto y la más costosa, la implementación. En esta fase se definirán las tecnologías a utilizar en el proyecto, así como las herramientas para trabajar con estas tecnologías.

Esto irá en función del análisis que hayamos realizado previamente, si el análisis ha sido bueno y se ha empleado el tiempo necesario en él, se utilizarán las tecnologías más adecuadas para implementar el proyecto, y la mejor manera para almacenar la información y trabajar con ella.

Por tanto, en esta fase es cuando el equipo empieza a desarrollar técnicamente la solución con las tecnologías elegidas y se coordina de forma efectiva para entregar la solución en el tiempo acordado.

Para esto, al finalizar esta fase de implementación se empezará con las labores de validación de las funcionalidades requeridas y las pruebas de todos los aspectos de la aplicación.

Aunque hay quién prefiere englobar la parte de validación y pruebas dentro de la fase de implementación, yo la he separado aquí para diferenciarla mejor y hacer una documentación más estructurada.

De esta manera pasaremos a explicar las diferentes decisiones que se han llevado en esta etapa en las siguientes páginas, y que son:

- Tecnologías y herramientas utilizadas.
- Implementación de funcionalidades.
- Algoritmo utilizado y proceso.

6.1 Tecnologías y herramientas utilizadas en la aplicación

A continuación se explican las distintas tecnologías utilizadas para el desarrollo de la aplicación, y también qué herramientas se han utilizado para implementar estas tecnologías.

Tecnologías utilizadas:

- **Framework de desarrollo .NET 4:** Esto es un componente integral que forma parte de Windows y que admite la compilación y generación de todo tipo de aplicaciones y servicios web. Es una plataforma enorme que está compuesta de muchas tecnologías que se explicarán más adelante. Las características principales más importantes de esta plataforma son: [referencia a pág. web]
 - Proporcionar un entorno coherente de programación orientada a objetos, en el que el código de los objetos se pueda almacenar y ejecutar de forma local, ejecutar de forma local pero distribuida en Internet o ejecutar de forma remota.
 - Proporcionar un entorno de ejecución de código que reduzca lo máximo posible la implementación de software y los conflictos de versiones.
 - Ofrecer un entorno de ejecución de código que promueva la ejecución segura del mismo, incluso del creado por terceras personas desconocidas o que no son de plena confianza.
 - Proporcionar un entorno de ejecución de código que elimine los problemas de rendimiento de los entornos en los que se utilizan scripts o intérpretes de comandos.
 - Ofrecer al programador una experiencia coherente entre tipos de aplicaciones muy diferentes, como las basadas en Windows o en el Web.
 - Basar toda la comunicación en estándares del sector para asegurar que el código de .NET Framework se puede integrar con otros tipos de código.

.NET Framework contiene dos componentes principales: Common Language Runtime y la biblioteca de clases de .NET Framework.

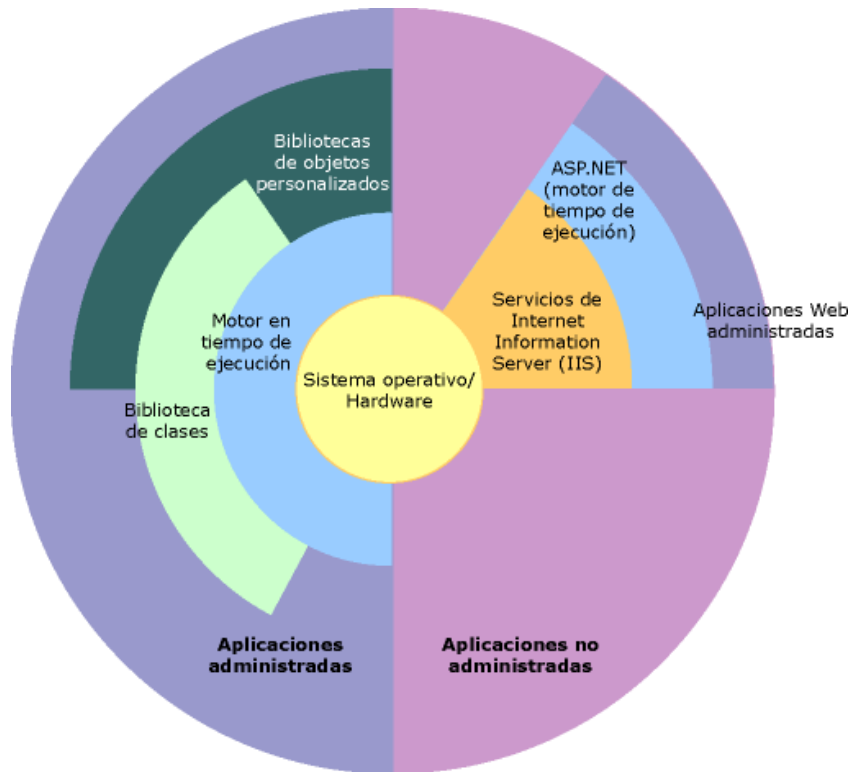


Ilustración 17: Arquitectura principal de la plataforma .NET

- **Common Language Runtime:** Este componente se separa aparte porque ofrece muchas características, aunque esté contenido en la enorme plataforma que es .NET Framework. Este componente ejecuta el código y proporciona servicios que facilitan el proceso de desarrollo.

El motor en tiempo de ejecución se puede considerar como un agente que administra el código en tiempo de ejecución y proporciona servicios centrales, como la administración de memoria, la administración de subprocesos y la comunicación remota, y que aplica una seguridad estricta a los tipos y otras formas de especificación del código que promueven su seguridad y solidez.

De hecho, el concepto de administración de código es un principio básico del motor en tiempo de ejecución. El código destinado al motor en tiempo de ejecución se denomina código administrado, a diferencia del resto de código, que se conoce como código no administrado.

- **Biblioteca de clases:** Se trata del otro componente principal de .NET Framework, es una completa colección orientada a objetos de tipos reutilizables que se pueden emplear para desarrollar aplicaciones que abarcan desde las tradicionales herramientas de interfaz gráfica de usuario (GUI) o de línea de comandos hasta las aplicaciones basadas en las innovaciones más recientes proporcionadas por ASP.NET, como los formularios Web y los servicios Web XML. Son todas las clases nativas contenidas dentro de .NET para todo tipo de funcionalidades.
- **Lenguaje de programación C#:** Dentro del conjunto de lenguajes de programación incluidos en la plataforma .NET se encuentra C# (C Sharp). Este ha sido el lenguaje utilizado para el código de la aplicación. Es un lenguaje fuertemente orientado a objetos, que utiliza una sintaxis parecida a Java y C++. Esta desarrollado y estandarizado por Microsoft como parte de su plataforma .NET, y que posteriormente fue aprobado como estándar por la ISO. Es un lenguaje simple, eficaz y con seguridad de tipos. Gracias a Visual Studio (Visual C#) permite desarrollar aplicaciones de forma rápida y eficiente gracias a que incorpora funcionalidades implementadas para que el programador no tenga que preocuparse de ciertos aspectos de programación, como componentes gráficos, control de eventos, cuadros de diálogo, gestión de hilos, etc.
- **Linq (Language Integrated Query):** Es una tecnología desarrollada por Microsoft que permite hacer todo tipo de consultas nativas semejantes a las de SQL a los lenguajes de la plataforma .NET. Esta tecnología define operadores de consulta estándar que permite a estos lenguajes filtrar, enumerar y crear proyecciones de gran variedad de tipos de colecciones de datos usando la misma sintaxis, por lo que hace mucho más simple la tarea, además de ser muy potente.

Entre estas colecciones soportadas se pueden encontrar arrays, listas, clases enumerables, tablas de bases de datos relacionales, etc.

Esta tecnología facilita mucho la labor al programador abstrayéndose de estas cuestiones de programación para que se dedique a implementar aquello que quiere hacer, y no preocupándose por el cómo.

El objetivo de esta tecnología es permitir que todo el código de estos lenguajes soportados esté orientado a objetos todo lo posible, ayudando al programador a poder abstraerse de la capa de persistencia de datos, entre otras cosas, como el tipo de colección que está utilizando.

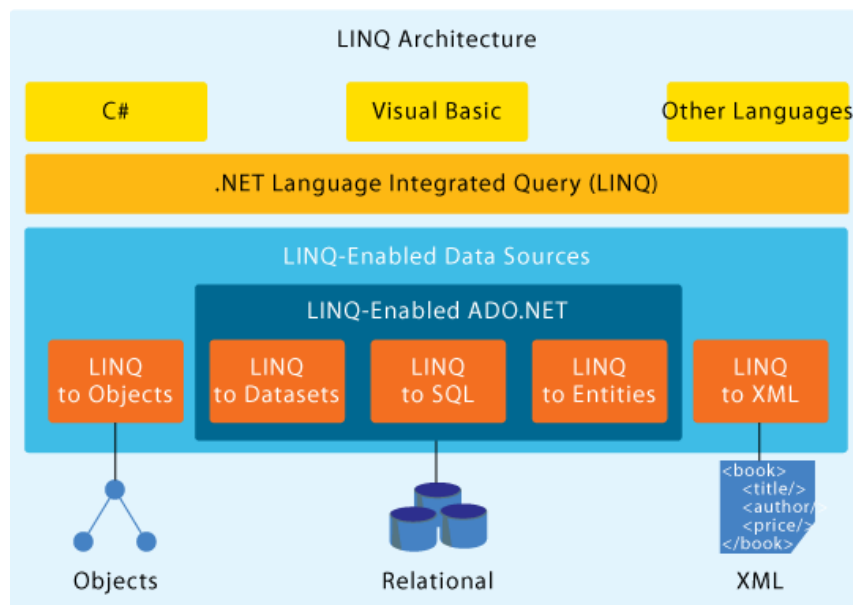


Ilustración 18: Arquitectura de la tecnología LINQ

- **Librería Interop.Excel:** Esta librería que viene incluida con el IDE de desarrollo Visual Studio, es una librería nativa de Microsoft para trabajar con la aplicación Excel. Con ella se pueden crear, abrir, modificar y leer ficheros Excel de tipo “.xls” o “.xlsx”. Se ha utilizado esta librería para trabajar con el fichero de entrada de la aplicación y para generar los distintos calendarios finales de exámenes en formato Excel.

- **Microsoft SQL Server Express 2008:** Este es el Sistema Gestor de Base de Datos. Se ha utilizado la versión Express porque para la base de datos que utiliza nuestra aplicación, que es bastante simple, nos es suficiente. Utiliza el modelo de las bases de datos relacionales. Como características se pueden destacar las siguientes: Soporta transacciones y procedimientos almacenados, es escalable, estable y seguro, y se integra muy bien con las aplicaciones desarrolladas en .NET. Esto último es lógico ya que también pertenece a Microsoft.

La base de datos utilizada en la aplicación se ha creado en este sistema.

- **ADO.NET y Entity Framework:** ADO.NET es un conjunto de componentes software incluidos con el .NET Framework. Sirven para acceder a datos y a servicios de datos. Se suele utilizar para acceder y modificar datos almacenados en Sistemas Gestores de Base de Datos, tanto relacionales como no relacionales. El Entity Framework es un conjunto de funciones y procedimientos de acceso a datos, esta implementado junto con ADO.NET. Se ven las entidades de la base de datos como objetos estandarizados donde se pueden ejecutar varios métodos para manipular esos datos.

Herramientas utilizadas:

- **Microsoft Visual Studio 2010 Professional:** Este es el IDE (Integrated Development Enviroment) o Entorno de Desarrollo Integrado, con el cual se ha desarrollado la aplicación. Es un IDE que consume una cantidad de recursos considerable, por lo que es algo pesado, pero que ofrece multitud de funcionalidades. Soporta gran cantidad de lenguajes de programación, y permite al programador crear aplicaciones de escritorio, sitios y aplicaciones web de muchos tipos, además de servicios web. De esta manera se pueden crear aplicaciones que se intercomuniquen entre estaciones de trabajo, páginas web y dispositivos móviles.

Aparte de tener un gran abanico de tipos de aplicaciones que se pueden crear simplifica la compilación, depuración y el despliegue de aplicaciones en gran variedad de plataformas como SharePoint y la Nube. Además dispone de soporte para pruebas y con herramientas de depuración de alta calidad.

- **Microsoft Office Visio 2007:** Esta herramienta permite el dibujar una gran variedad de tipos de diagramas. Tiene varias categorías como diagramas de flujo, diagramas de empresa, diagramas de ingeniería, diagramas de software y base de datos, etc. Dentro de estas categorías hay bastantes diagramas para elegir, por lo que es una aplicación bastante completa. Se ha utilizado esta herramienta para desarrollar el Diagrama de Casos de Uso, el Diagrama de Componentes y el Diagrama de Clases de la aplicación.
- **Microsoft Office Project 2007:** Es una aplicación destinada a la creación, la organización, la planificación y el seguimiento de proyectos. No está enfocada a los proyectos software, por lo tanto se puede utilizar para cualquier tipo genérico de proyectos, incluidos los de software. La herramienta permite definir los recursos para el proyecto, definir calendarios y crear diagramas de planificación y seguimiento.
El diagrama de Gantt y los presupuestos se han realizado con esta herramienta.
- **Microsoft Office Excel 2007:** Aplicación destinada a crear hojas de cálculo, además de otras muchas cosas, como crear gráficos a partir de esas hojas. Se utiliza normalmente para tareas financieras y contables, pero de forma genérica se trata de una aplicación que permite almacenar, consultar y manipular datos numéricos y alfanuméricos dispuestos en forma de tablas compuestas por celdas. En una celda (que es la unidad básica de información) se insertan los valores y las fórmulas que realizan los cálculos, además se pueden utilizar funciones y crear distintos tipos de gráficos.
Se ha utilizado este programa para crear el fichero de entrada de la aplicación con todas las asignaturas incluidas las de ADE, ya que nos permite validar datos de forma sencilla y que el usuario no pueda introducir datos incorrectos de asignaturas.

- **SQL Server Management Studio:** Herramienta gráfica que se puede utilizar para crear y administrar bases de datos de SQL Server. Puede realizar todas las tareas de administración de una base de datos relacional, y también permite crear diagramas relacionales, entre otras muchas cosas. Ha sido la herramienta utilizada para ejecutar diferentes consultas a la base de datos y poder comprobar ciertos aspectos de programación para la aplicación, también se ha utilizado para crear el diagrama Entidad Relación (aunque no esté dibujado exactamente igual que un diagrama E/R puro).

Como conclusión de este apartado hay que decir que se han podido utilizar estas tecnologías y herramientas porque la Universidad cuenta con licencias de software para ellas, ya que no son de libre distribución y además son caras, porque están dirigidas a las empresas.

También existen versiones Express que son libres y no tienen coste, pero también tienen menos funcionalidades. El caso es que se han elegido las versiones completas para investigar más con su uso y porque así se aprovechan las licencias que ha comprado la Universidad. La versión de SQL Server sí que es la versión Express, que es gratis, ya que nuestra base de datos es muy pequeña y no necesitamos una versión mas completa de SQL Server.

Las razones técnicas de usar estas tecnologías y no otras, que puedan ser de libre distribución por ejemplo, se han descrito anteriormente en el apartado “3.4 Estudio de viabilidad”.

6.2 Implementación de las funcionalidades

En este apartado describiremos como se han implementado las distintas funcionalidades de la aplicación.

Se explicará cómo se ha desarrollado todo el código de la aplicación y los distintos módulos en los que se ha separado (capas), describiendo la funcionalidad de cada componente y de cada clase.

También se explicará la implementación de la base de datos, qué es lo que significa cada campo y que valores podrá almacenar.

Empezaremos describiendo la implementación de la base de datos, para luego pasar a la implementación del código de la aplicación.

Base de Datos

El nombre que le hemos puesto a la Base de Datos ha sido “CalendarioExámenes”. La base de datos no contiene vistas, ni procedimientos, ni funciones, ni desencadenadores de ningún tipo, etc. Solo contiene tablas ya que es una base de datos que tiene un diseño sencillo, y por tanto no son necesarias estas funcionalidades en la base de datos.

En total hay cinco tablas para almacenar los datos necesarios. A continuación se describirá cada una de las tablas:

Nombre de tabla	“Año”		
Base de datos	CalendarioExámenes		
Claves primarias	“Id”		
Descripción	En esta tabla se almacenará el año actual al que hacen referencia las asignaturas que hay en la base de datos. Es como una base de datos que sirve de control para la aplicación, para saber si se necesita importar nuevos datos de asignaturas del fichero de entrada o no.		
Columnas			
Nombre	Tipo de datos	Propiedades	Descripción
Id	Entero	Requerido	Valor numérico que identifica unívocamente a un año.
Descripción	Texto(50)	Requerido	Texto que nos indica el año actual, en la forma Año n / n + 1

Ilustración 19: Tabla “Año”

Nombre de tabla	“Asignatura”		
Base de datos	CalendarioExámenes		
Claves primarias	“Código”, “IdTitulacion”, “IdCurso”		
Descripción	En esta tabla se guardan todos los datos de las diferentes asignaturas de las que constan las titulaciones.		
Columnas			
Nombre	Tipo de datos	Propiedades	Descripción
Código	Numérico	Requerido. Entero	Valor numérico de cinco dígitos que identifica “parcialmente” a una asignatura, ya que hay casos excepciones que con el código no vale.
IdTitulacion	Numérico	Requerido. Entero	Clave externa: Valor numérico que identifica unívocamente una Titulación.
IdCurso	Numérico	Requerido. Entero[1,6]	Clave externa: Valor numérico que identifica unívocamente un Curso.
IdCuatrimestre	Numérico	Requerido. Entero[1,2]	Clave externa: Valor numérico que identifica unívocamente un Cuatrimestre.
Nombre	Texto(100)	Requerido	Nombre de la asignatura.
Especialidad	Texto(100)	Requerido	Especialidad de la asignatura (mención).
Subtitulación	Texto(100)	Requerido. Por defecto NO	Almacenará la titulación nativa de una asignatura de un Doble Grado.
Optativa	Bit	Requerida. Por defecto 0 (False)	Nos dirá si la asignatura es optativa o no.
FechaExamen	DateTime	Opcional. Por defecto NULL	Almacenará la fecha de examen de la asignatura.
FechaExtraordinaria	DateTime	Opcional. Por defecto NULL	Almacenará la fecha del examen extraordinario de la asignatura.
Peso	Numérico	Requerido. Entero[1,2]	Almacenará la dificultad de la asignatura. 1 = Normal. 2 = Difícil.

Ilustración 20: Tabla “Asignatura”

Nombre de tabla	“Cuatrimestre”		
Base de datos	CalendarioExamenes		
Claves primarias	“Id”		
Descripción	En esta tabla se guardan los datos relativos a los cuatrimestres que hay. Como solo hay dos cuatrimestres, contendrá dos registros		
Columnas			
Nombre	Tipo de datos	Propiedades	Descripción
Id	Numérico	Requerido. Entero[1,2]	Valor numérico que identifica unívocamente a un cuatrimestre.
Nombre	Texto(50)	Requerido	Nombre del cuatrimestre. Ej. “Primer Cuatrimestre”

Ilustración 21: Tabla “Cuatrimestre”

Nombre de tabla	“Curso”		
Base de datos	CalendarioExámenes		
Claves primarias	“Id”		
Descripción	En esta tabla se guardan los datos relativos a los distintos cursos que hay.		
Columnas			
Nombre	Tipo de datos	Propiedades	Descripción
Id	Numérico	Requerido. Entero[1,6]	Valor numérico que identifica unívocamente a un curso. En principio habrá hasta seis cursos, ya que los Dobles Grados tienen seis.
Nombre	Texto(50)	Requerido	Nombre del curso. Ej. “Segundo Curso”.

Ilustración 22: Tabla “Curso”

Nombre de tabla	“Titulación”		
Base de datos	CalendarioExámenes		
Claves primarias	“Id”		
Descripción	En esta tabla se guardan los datos relativos a las dos titulaciones que hay (puede que un futuro haya más).		
Columnas			
Nombre	Tipo de datos	Propiedades	Descripción
Id	N Numérico	Requerido. Entero	Valor numérico que identifica unívocamente a una titulación.
Nombre	Texto(50)	Requerido	Nombre de la titulación.

Ilustración 23: Tabla “Titulación”

Funcionalidades del software:

A continuación describiremos las funcionalidades que realiza cada componente de nuestra aplicación software. Como ya se ha dicho anteriormente la aplicación consta de tres componentes:

- Dominio (Capa de datos).
- GUI (Capa de presentación).
- LogicaNegocio (Capa del controlador).

El componente Dominio se encarga de definir los datos necesarios para la generación de los calendarios de exámenes.

El componente GUI se encarga de definir los formularios gráficos con los que el usuario podrá interactuar con la aplicación.

El componente LogicaNegocio se encarga de definir la lógica para que la aplicación haga lo que el usuario quiere hacer.

Componente Dominio: Este componente contiene tres clases, donde cada una tendrá unas funciones específicas:

- Clase Constantes: Esta clase estática almacenará datos constantes utilizados en toda la aplicación. Se encapsulan aquí por motivos de eficiencia, para que estos datos solo estén en un sitio.

- Clase Nodo: Esta clase contendrá la definición de un nodo del algoritmo. Un nodo estará compuesto por un día, una lista de asignaturas para el turno de mañana de ese nodo, y otra lista de asignaturas para el turno de tarde de ese nodo. Ambas podrán estar vacías.

»

- Clase Valor: Esta clase almacenará los distintos valores (puntuaciones) que existirán para cada turno del día de una asignatura. Hay que decir que cuanto más alto sean estos valores, más difícil será colocar la asignatura en ese turno.

»

Componente GUI: Este componente contiene cinco clases, que son clases de formularios, es decir, todas las pantallas de las que está compuesta la aplicación y con las que el usuario podrá interactuar. Estas clases serán más extensas porque también contendrán todos los eventos que se disparen en los formularios, además de los controles (cajas de texto, etiquetas, etc.):

- Clase Formulario “FrmMensaje”: Esta clase contendrá un formulario de tipo mensaje que se mostrará cuando se estén cargando los datos del fichero de entrada, con el objetivo de informar al usuario de que se está realizando esa operación y debe esperar.

- Clase Formulario “FrmElegirOptativas”: Esta clase contendrá el formulario para que el usuario pueda elegir las asignaturas optativas que se vayan a impartir el año actual, si lo desea. También podrá deseleccionarlas si lo desea.

□

- Clase Formulario “FrmMostrarAsignaturas”: Esta clase contendrá el formulario para que el usuario pueda visualizar unas asignaturas por criterio. Actualmente hay tres criterios: Mostrar las asignaturas de Informática, las del Doble Grado en Informática – ADE, y mostrar las asignaturas optativas que se van a tener en cuenta para la generación del calendario.

¶

- Clase Formulario “FrmPrevisualización”: Clase que contendrá el formulario de previsualización del calendario generado, que se mostrará después de que el usuario haya generado el calendario con las opciones que haya elegido.

□

- Clase Formulario “FrmPrincipal”: Esta clase contendrá el formulario principal de la aplicación, es decir, la ventana principal donde el usuario podrá elegir todas las opciones para la generación del calendario, como las fechas, las asignaturas optativas, el algoritmo, etc. Esta clase tiene una gran cantidad de variables y métodos por ser el formulario principal donde se encuentran la mayoría de acciones que el usuario puede realizar, por lo tanto solo se mostrarán los métodos que utiliza.

¶

Componente LogicaNegocio: Este componente contiene las clases necesarias para implementar la lógica de la aplicación, como por ejemplo los métodos relacionados con el algoritmo de generación.

- Clase “TrabajarAlgoritmo”: Esta clase contendrá los métodos para implementar los algoritmos de generación del calendario.

11

- Clase “TrabajarAsignaturaSQL”: Clase que contendrá toda la lógica relacionada con el acceso y gestión de las asignaturas contenidas de la base de datos.

12

- Clase “TrabajarExcel”: Esta clase contendrá todos los métodos necesarios para poder implementar las funcionalidades que queremos con la aplicación Excel, es decir, para leer nuestro fichero de entrada, y también para exportar los resultados del calendario a ficheros Excel.

11

Ficheros de entrada y salida de la aplicación:

Por último, para concluir con este subapartado hay que decir que la forma de almacenamiento elegida, tanto para los datos de entrada de las asignaturas, como para los calendarios de exámenes generados, han sido ficheros formateados de tipo Excel. La razón de esta elección ha sido porque son más prácticos para el usuario, y puede visualizar los datos de las asignaturas de una manera clara, sencilla y elegante.

Además, para el caso del fichero de entrada permite hacer validaciones de datos lo que nos viene de maravilla para nuestro propósito, y para el caso del fichero de salida permite el poder exportarlo a un PDF, o imprimirlo, y queda muy presentable.

6.3 Proceso del algoritmo utilizado

En este apartado explicaremos el algoritmo que se ha utilizado para colocar los exámenes de las asignaturas en la franja de días que nos dan.

Este algoritmo debe de trabajar con una serie de restricciones como hemos comentado anteriormente, que tiene que cumplir siempre que se pueda. Lo que ocurre es que infringir una de estas restricciones puede ser peor que infringir otra, por lo que hay prioridades.

Para empezar definiremos todas estas prioridades del algoritmo. Estas prioridades ya están descritas en el apartado 3.2, ya que también entran dentro de los requisitos de usuario, pero se volverán a definir aquí porque es necesario para entender el algoritmo mejor y porque también definiremos el “daño” que tiene cada restricción si se incumple. El daño que tendría el colocar una asignatura X en un turno determinado Y será distinto para cada restricción, dependiendo de la prioridad de esta, y se ha definido como:

(“Valor de daño de la restricción”) x “Multiplicador”.

Donde multiplicador se define como el número de asignaturas que incumplirían esa restricción si quisiéramos insertar la asignatura X en el turno Y correspondiente.

El conjunto total de restricciones es el siguiente (El valor numérico de los daños los definiremos por letras):

- Restricción 1) En días adyacentes del calendario no puede haber exámenes de asignaturas que sean del mismo curso y especialidad. Daño = “p” (x mult).
- Restricción 2) No puede haber exámenes de asignaturas del mismo curso y misma especialidad en el mismo día y turno del calendario. Daño = “q” (x mult).
- Restricción 3) En el mismo día y en el mismo turno no puede haber exámenes de asignaturas que sean de cursos consecutivos y misma especialidad. Daño = “r” (x mult).
- Restricción 4) No puede haber exámenes de asignaturas del mismo curso y especialidad en el mismo día, pero en distinto turno. Daño = “s” (x mult).
- Restricción 5) No puede haber exámenes de asignaturas de cursos consecutivos y misma especialidad el mismo día pero distinto turno del calendario. Daño = “t” (x mult).

- Restricción 6) Las asignaturas con dificultad alta tienen que estar separadas lo máximo posible unas de otras. Daño = “u” (x mult).
- Restricción 7) Es preferible que los exámenes sean colocados en el turno de mañana, siempre que sea posible. (No tiene daño).

Hay que comentar las dos últimas restricciones porque tienen un tratamiento distinto para calcularse. En cuanto al daño que sufre el incumplimiento de la restricción de la dificultad de las asignaturas hay que decir que hay varios multiplicadores dependiendo del día en el que haya asignaturas que rompen la restricción:

- Multiplicador = 5 → Hay una asignatura Y de dificultad alta ese mismo día donde queremos insertar una asignatura X.
- Multiplicador = 4 → Hay una asignatura Y de dificultad alta un día adyacente de donde queremos insertar una asignatura X.
- Multiplicador = 3 → Hay una asignatura Y de dificultad alta dos días adyacentes donde queremos insertar una asignatura X.
- Multiplicador = 2 → Hay una asignatura Y de dificultad alta tres días adyacentes donde queremos insertar una asignatura X.
- Multiplicador = 1 → Hay una asignatura Y de dificultad alta cuatro días adyacentes donde queremos insertar una asignatura X.

En cuanto a la última restricción descrita no tiene daño, ya que lo que se hará será insertar una asignatura por la mañana si hay algún valor para algún turno de mañana que sea menor (menos daño) que cualquier valor para un turno de tarde, de esa manera nos aseguramos de cumplir esa restricción.

En cuanto al algoritmo utilizado hay que decir que se ha utilizado un algoritmo englobado dentro de los métodos de búsqueda heurística de la Inteligencia Artificial. Tendremos una función de evaluación, que es la que hemos definido más arriba, y cuanto mayor sea esta función para un nodo en un momento dado más difícil será insertarlo en el turno del día dado.

Se ha elegido un algoritmo muy parecido a los que están dentro de las Técnicas de Búsqueda Local, por las razones siguientes:

- Este tipo de algoritmos se usan cuando el camino hasta el objetivo es irrelevante y solamente interesa el estado objetivo, por lo que se ajusta mucho a la descripción de nuestro problema.
- Se utilizan para resolver problemas de optimización puros, en los que no existe un estado objetivo, sino que se trata de encontrar el mejor estado según una función objetivo. De nuevo esto se ajusta perfectamente a las características de nuestro problema.
- Además estos algoritmos tienen unas ventajas características, como por ejemplo que usan menos memoria porque no necesitan mantener los caminos que se han ido desarrollando hasta la solución.

Por estas razones se ha desarrollado un algoritmo parecido a los de este tipo similar al llamado Búsqueda en Escalada (Hill-Climbing), el cual es un algoritmo voraz que no mantiene un árbol de búsqueda y que no mira más allá de los vecinos inmediatos del nodo actual. En cada paso elige el nodo vecino que tiene un mejor valor de la función objetivo. En este algoritmo la búsqueda se puede quedar atascada en un máximo local o un mínimo local.

Por eso nuestro algoritmo no es un algoritmo Hill-Climbing puro porque nuestro algoritmo si es capaz de mirar todos los nodos del problema, sean vecinos o no, aunque no tenga que mantener todos los caminos, solo los valores de la función en todos los nodos en el momento actual. Se podría decir que nuestro algoritmo es una mezcla entre la Búsqueda en Escalada y las Técnicas de Mejor-Primero de la Inteligencia Artificial.

Esto hace que el algoritmo no se pueda quedar atascado nunca en un máximo o mínimo local, ya que tenemos en memoria el valor de todos los nodos, no solo de los vecinos inmediatos. Otra ventaja de utilizar este algoritmo es que permite encontrar soluciones en espacios de tiempo muy cortos. Lo óptimo que sea depende por tanto de la calidad de la función de evaluación que le asignemos y de cómo lo llevemos a cabo.

Para concluir este apartado sobre el algoritmo utilizado definiremos el proceso que sigue el algoritmo de forma genérica, para entenderlo mejor. Los pasos que dará el algoritmo son los siguientes:

- 1) Se ejecutará un ciclo del algoritmo con la lista de asignaturas del Doble Grado. Esto se hace por la razón de que es preferible asignar primero las asignaturas compartidas entre la titulación del Grado en Informática y la del Doble Grado en Informática – ADE.
- 2) Se colocarán primero los exámenes de las asignaturas que nos vienen dadas ya desde el principio, que son las de ADE.
- 3) Se ordenará la lista de asignaturas por curso, para ir asignando primero las asignaturas de primer curso, segundo curso, etc.
- 4) Se irán colocando los exámenes de las asignaturas recorriendo curso por curso, hasta que tengamos todas las asignaturas colocadas en los huecos que nos van quedando.
- 5) Cuando se hayan colocado todas repetiremos el ciclo pero esta vez con las asignaturas restantes del Grado en Informática. Al decir restantes es porque tendremos que eliminar las asignaturas ya asignadas compartidas, teniendo especial cuidado en que en este ciclo esas asignaturas ya asignadas habrá que tener en cuenta el curso que tienen en la titulación de Informática y no el del Doble Grado (ya que estos cursos en la mayoría de los casos serán distintos).
- 6) Una vez se hayan colocado los exámenes de todas las asignaturas en el rango de días que nos dan, se devuelve una lista de nodos con todas las asignaturas ya colocadas.

7. Validación y pruebas

Aquí definiremos todas las pruebas de validación (pruebas de usuario), y veremos si se obtienen los resultados correctamente.

Identificaremos las pruebas de validación numéricamente del modo “Prueba-n”, donde “n” será un número que empiece en 1 y se vaya sumando uno a uno. También pondremos en qué consiste la prueba (descripción).

Estas pruebas tienen que verificar el funcionamiento correcto de los requisitos de usuario definidos en el listado de requisitos, por tanto también pondremos que requisito es el que cumple, o requisitos, y que especificaciones tiene.

Por tanto, usaremos la siguiente plantilla para especificar las pruebas de usuario:

[Identificador de la prueba]	
Nombre	[Nombre de la prueba]
Objetivo	[Objetivo de la prueba]
Requisito de usuario	[Requisito/s que cumple la prueba]
Especificaciones de entrada	[Especificaciones de entrada de la prueba]
Especificaciones de salida	[Especificaciones de salida de la prueba]
Resultado	[Resultado de la prueba]

Ilustración 24: Plantilla de prueba de usuario

7.1 Pruebas de usuario

PRUEBA-01	
Nombre	Prueba Importar
Objetivo	Verificar que el usuario puede importar los datos del fichero de entrada
Requisito de usuario	RU-10
Especificaciones de entrada	El usuario deberá encontrarse en el primer paso de la pantalla principal de la aplicación, y pinchar en el botón “Importar asignaturas”
Especificaciones de salida	La aplicación muestra un cuadro de diálogo para elegir el fichero Excel y un mensaje cuando termine de procesarlo
Resultado	OK

PRUEBA-02	
Nombre	Prueba Exámenes ADE
Objetivo	Verificar que todas las asignaturas de la rama de ADE tienen fecha de examen puesta
Requisito de usuario	RU-09
Especificaciones de entrada	El usuario deberá encontrarse en el primer paso de la pantalla principal de la aplicación, y pinchar en el botón “Mostrar asignaturas Doble Grado”
Especificaciones de salida	La aplicación muestra todas las asignaturas del Doble Grado, y entonces el usuario podrá comprobar que todas las asignaturas de ADE tienen fecha de examen asignada
Resultado	OK

PRUEBA-03	
Nombre	Prueba Elección Optativas
Objetivo	Verificar que el usuario puede elegir las optativas que quiera para el calendario de exámenes
Requisito de usuario	RU-11, RU-12
Especificaciones de entrada	El usuario deberá encontrarse en el primer paso de la pantalla principal de la aplicación, y pinchar en el botón “Elegir optativas”
Especificaciones de salida	La aplicación muestra todas las asignaturas optativas del Grado en Informática y el usuario podrá elegir las que desee. Si luego pincha en el botón “Mostrar optativas seleccionadas” podrá comprobar cuales ha elegido
Resultado	OK

PRUEBA-04	
Nombre	Prueba Mostrar asignaturas
Objetivo	Verificar que el usuario puede ver todas las asignaturas de las dos titulaciones que se tendrán en cuenta para la generación del calendario
Requisito de usuario	RU-13
Especificaciones de entrada	El usuario deberá encontrarse en el primer paso de la pantalla principal de la aplicación, y pinchar en el botón “Mostrar asignaturas Informática” o “Mostrar asignaturas Doble Grado”
Especificaciones de salida	La aplicación muestra todas las asignaturas del Grado en Informática en el primer caso, y todas las asignaturas del Doble Grado en el segundo
Resultado	OK

PRUEBA-05	
Nombre	Prueba Elección período
Objetivo	Verificar que el usuario puede elegir el período a generar en el calendario de exámenes: Primer cuatrimestre, segundo, o convocatoria extraordinaria
Requisito de usuario	RU-14
Especificaciones de entrada	El usuario deberá encontrarse en el segundo paso de la pantalla principal de la aplicación, y pinchar en el período deseado
Especificaciones de salida	La aplicación muestra la fecha de inicio del calendario y la fecha de fin, para que el usuario pueda elegir las, según el período elegido
Resultado	OK

PRUEBA-06	
Nombre	Prueba Elección fechas
Objetivo	Verificar que el usuario puede elegir las fechas de inicio y de fin del calendario de exámenes a generar
Requisito de usuario	RU-01, RU-02
Especificaciones de entrada	El usuario deberá encontrarse en el segundo paso de la pantalla principal de la aplicación, y pinchar en el período deseado. A continuación elegirá una fecha de inicio del calendario y una fecha de fin
Especificaciones de salida	La aplicación muestra las fechas de inicio y de fin elegidas por el usuario
Resultado	OK

PRUEBA-07	
Nombre	Prueba Elección sábados
Objetivo	Verificar que el usuario puede elegir que los días que sean sábado no se tengan en cuenta en el calendario de exámenes a generar
Requisito de usuario	RU-15
Especificaciones de entrada	El usuario deberá encontrarse en el segundo paso de la pantalla principal de la aplicación, y pinchar en la caja de selección “No contar con sábados”, para dejarla seleccionada
Especificaciones de salida	Cuando se haya generado el fichero de salida del calendario de exámenes, el usuario podrá comprobar que efectivamente no hay sábados en el calendario
Resultado	OK

PRUEBA-08	
Nombre	Prueba Turnos
Objetivo	Verificar que cada día del calendario de exámenes tiene dos turnos: el de mañana, y el de tarde
Requisito de usuario	RU-17
Especificaciones de entrada	Ninguna
Especificaciones de salida	Cuando se haya generado el fichero de salida del calendario de exámenes, el usuario podrá comprobar que cada día en el calendario tiene el turno de mañana y el turno de tarde
Resultado	OK

PRUEBA-09	
Nombre	Prueba domingos
Objetivo	Verificar que los días que sean domingo no deberán tenerse en cuenta en el calendario de exámenes en ningún caso
Requisito de usuario	RU-18
Especificaciones de entrada	Ninguna
Especificaciones de salida	Cuando se haya generado el fichero de salida del calendario de exámenes, el usuario podrá comprobar no hay domingos en el calendario
Resultado	OK

PRUEBA-10	
Nombre	Prueba de la validación de los datos de entrada
Objetivo	Verificar que el usuario no pueda introducir valores que no sean válidos en el fichero de entrada de datos de asignaturas
Requisito de usuario	RU-20
Especificaciones de entrada	El usuario deberá tener abierto el fichero Excel de entrada con los datos de las asignaturas.
Especificaciones de salida	Se comprueba que el usuario debe elegir de una lista desplegable los distintos datos de las asignaturas en el fichero Excel
Resultado	OK

PRUEBA-11	
Nombre	Prueba de turnos para sábados
Objetivo	Verificar que los días que sean sábado solo pueden tener un turno: el turno de mañana
Requisito de usuario	RU-21
Especificaciones de entrada	El usuario deberá haber elegido en el segundo paso de la aplicación la opción de contar con los sábados en el calendario
Especificaciones de salida	Cuando se haya generado el fichero de salida del calendario de exámenes, el usuario podrá comprobar que los días que sean sábado solo tienen el turno de mañana
Resultado	OK

PRUEBA-12	
Nombre	Prueba de almacenamiento calendario
Objetivo	Verificar que el calendario de exámenes generado se guarda en un fichero de salida Excel para que el usuario pueda consultarlo
Requisito de usuario	RU-31
Especificaciones de entrada	El usuario deberá estar en el paso cuarto de la aplicación y pinchar en el botón "Exportar"
Especificaciones de salida	El usuario podrá comprobar que se crea un fichero Excel con el calendario de exámenes generado en una carpeta del disco C
Resultado	OK

PRUEBA-13	
Nombre	Prueba de almacenamiento calendario
Objetivo	Verificar que el usuario puede elegir un curso y una titulación para exportar el calendario al fichero Excel
Requisito de usuario	RU-33
Especificaciones de entrada	El usuario deberá estar en el paso cuarto de la aplicación y seleccionar la titulación deseada, y a continuación el curso deseado para exportar el calendario
Especificaciones de salida	El usuario podrá comprobar que se crea un fichero Excel con el calendario de exámenes generado y que ese calendario solo es para esa titulación y ese curso que el usuario ha elegido anteriormente
Resultado	OK

7.2 Pruebas de rendimiento

En este apartado se analizarán los aspectos de rendimiento de la aplicación en cuanto a la generación del calendario de exámenes. Para medir la eficacia del proceso de generación tendremos en cuenta los tiempos que tarda, así como los resultados obtenidos si cambiamos prioridades en las restricciones del algoritmo.

Como hemos hablado en el apartado 6.3 “Proceso del algoritmo utilizado”, el algoritmo debe tratar una serie de restricciones donde cada restricción es más o menos prioritaria que otra.

Al implementar estas prioridades con valores numéricos, tenemos la oportunidad de comprobar la eficacia del algoritmo cambiando estas prioridades, para ver los distintos resultados que se pueden obtener y compararlos.

De esta manera, si comprobamos que con una configuración determinada de prioridades obtenemos resultados donde alguna restricción importante se incumple varias veces, podemos cambiar esos valores numéricos para las prioridades y comparar los resultados, para ver cómo se cumplen las distintas restricciones, y asegurarnos que las que son importantes se cumplen siempre que sea posible.

Hay que dejar claro que este apartado se centra solo en la eficacia del proceso, ya que en nuestra aplicación el rendimiento de la carga del fichero de entrada a la base de datos es algo irrelevante, porque básicamente el tiempo consumido en esta operación se debe a las operaciones de la librería dedicada al manejo del Excel. En nuestro fichero de entrada, el cual contiene los datos de todas las asignaturas de las dos titulaciones este tiempo de carga del fichero a la base de datos es constante, y se sitúa en los cinco segundos.

El tiempo que tarda en generar el fichero Excel de salida donde se escribe en calendario de exámenes también es algo irrelevante, por las mismas razones expuestas anteriormente. Como datos simplemente diremos que el tiempo máximo de este proceso es de alrededor de dos segundos, pero esto tampoco nos interesa medirlo porque también es muy pequeño y es constante para las dos titulaciones.

Por tanto, vamos a medir la eficacia en el proceso de generación del calendario por el algoritmo utilizado. Trataremos dos aspectos: el tiempo utilizado en la ejecución del proceso de generación y la eficacia del cumplimiento de las restricciones.

Tiempos de generación

En la siguiente tabla podemos ver los tiempos de diez generaciones de calendarios diferentes para cada período elegido, ya que la generación tardará más o menos tiempo en función del tipo de período elegido, porque en cada período hay un distinto número de asignaturas.

Para la generación de calendarios de exámenes del primer cuatrimestre se han hecho diez iteraciones, para poder hacer una mejor medición, con rangos de doce días, obteniendo los siguientes resultados:

Primer Cuatrimestre	
Iteración 1	18630 milisegundos
Iteración 2	16329 milisegundos
Iteración 3	18413 milisegundos
Iteración 4	19573 milisegundos
Iteración 5	18357 milisegundos
Iteración 6	17757 milisegundos
Iteración 7	17647 milisegundos
Iteración 8	17277 milisegundos
Iteración 9	17693 milisegundos
Iteración 10	18158 milisegundos

Ilustración 25: Tiempos generación primer cuatrimestre

Por tanto, podemos concluir con que la generación del calendario de exámenes para el primer cuatrimestre tiene una duración media de 17983.4 milisegundos, o lo que es lo mismo, 17.98 segundos.

Para la generación de calendarios de exámenes del segundo cuatrimestre se han hecho diez iteraciones, para poder hacer una mejor medición, con rangos de doce días, obteniendo los siguientes resultados:

Segundo Cuatrimestre	
Iteración 1	1094 milisegundos
Iteración 2	1000 milisegundos
Iteración 3	1023 milisegundos
Iteración 4	1016 milisegundos
Iteración 5	1064 milisegundos
Iteración 6	1037 milisegundos
Iteración 7	974 milisegundos
Iteración 8	1041 milisegundos
Iteración 9	977 milisegundos
Iteración 10	1051 milisegundos

Ilustración 26: Tiempos generación segundo cuatrimestre

Por tanto, podemos concluir con que la generación del calendario de exámenes para el segundo cuatrimestre tiene una duración media de 1027.7 milisegundos, o lo que es lo mismo, 1.03 segundos.

Como vemos el tiempo de la generación del segundo cuatrimestre es muy inferior al tiempo de generación del primer cuatrimestre. Esto es así porque en el segundo cuatrimestre tenemos muchas menos asignaturas a asignar que en el primer cuatrimestre, porque la mayoría son compartidas entre las dos titulaciones.

Para la generación de calendarios de exámenes de la convocatoria extraordinaria se han hecho diez iteraciones, para poder hacer una mejor medición, con rangos de doce días, obteniendo los siguientes resultados:

Convocatoria Extraordinaria	
Iteración 1	36435 milisegundos
Iteración 2	36947 milisegundos
Iteración 3	36119 milisegundos
Iteración 4	37040 milisegundos
Iteración 5	36667 milisegundos
Iteración 6	36993 milisegundos
Iteración 7	36629 milisegundos
Iteración 8	37089 milisegundos
Iteración 9	36580 milisegundos
Iteración 10	36141 milisegundos

Ilustración 27: Tiempos generación convocatoria extraordinaria

Por tanto, podemos concluir con que la generación del calendario de exámenes para la convocatoria extraordinaria tiene una duración media de 36664 milisegundos, o lo que es lo mismo, 36.6 segundos.

Podemos observar como los tiempos de generación de la convocatoria extraordinaria son bastante mayores a los tiempos de los otros dos períodos, lo que también es completamente normal, ya que en este período hay que asignar un número de asignaturas mucho mayor.

Como conclusión podemos decir que los tiempos de generación del calendario de exámenes gracias a nuestro algoritmo utilizado son muy buenos, ya que los mayores no sobrepasan los treinta y seis segundos, lo cual es un tiempo muy bueno en relación con el número de exámenes de asignaturas que hay que colocar en la franja del calendario, ya que el usuario apenas tiene que esperar a que termine la generación.

Eficacia en las restricciones

Medir la calidad de los calendarios de exámenes generados es una tarea difícil, ya que en una franja de doce días siempre va a haber alguna restricción que alguna vez se incumplirá debido al gran número de asignaturas que colocar para un período. Lo que se pretende conseguir es que las restricciones que se incumplan sean las que hagan menos “daño”, como se ha explicado anteriormente, en el proceso del algoritmo.

Por tanto, para medir esta eficiencia en la colocación de los exámenes de las asignaturas en el calendario lo que haremos será generar un espectro amplio de calendarios de exámenes, pero cambiando los valores numéricos que identifican al “daño” que hace que cada restricción se incumpla.

De esta manera, podremos comparar los distintos resultados obtenidos, comparando las restricciones que se incumplen, y el número de veces que esto ocurre con cada restricción. Por cada prueba que hagamos, cambiaremos las prioridades de estas restricciones para comparar nuestros calendarios generados y poder llegar a una solución donde los calendarios generados sean lo más óptimos posible.

Para la realización de estas pruebas tendremos en cuenta que se harán siempre en el peor de los casos, por ejemplo, teniendo en cuenta todas las asignaturas optativas de Informática como si se fueran a impartir para el curso actual; además también depende de cómo nos vengán impuestas las fechas de los exámenes de la rama de ADE, que ya serán fijas desde el principio y son inamovibles.

También se tendrá en cuenta que la franja de días para el calendario de exámenes será de los doce días por defecto, incluyendo los dos sábados, que es como se está realizando actualmente la organización de los calendarios de exámenes de la Universidad, y se tomará como referencia el calendario exportado para todos los cursos de la titulación a mostrar.

También hay que fijarse en que los resultados respecto a la dificultad de las asignaturas son complicados, ya que si hay varias asignaturas difíciles en un cuatrimestre y tienen que estar colocadas en un rango de doce días es difícil que no estén juntas (al mínimo una tendrá que estar junta, y las otras con solo dos días de separación), por lo que estos resultados son relativos, dependiendo de las asignaturas que el usuario considere difíciles al hacer la carga del fichero de entrada, es un punto importante a tener en cuenta, y estos resultados se agravarán en el caso de la convocatoria extraordinaria.

Por tanto, las pruebas las haremos en referencia a los calendarios generados para el primer cuatrimestre, ya que es el período que mayor número de asignaturas tiene, sin contar con la convocatoria extraordinaria (en el que están todas). En la convocatoria extraordinaria habrá muchas restricciones que se incumplan más de una vez, pero esto es completamente normal porque, como hemos dicho, doce días es un período muy reducido para intentar cumplir la mayoría de las restricciones por la gran cantidad de asignaturas que hay.

Así pues empezaremos definiendo las pruebas con los valores iniciales para las prioridades de cada restricción que se tuvieron en cuenta, teniendo en cuenta que unas eran más importantes que otras. Se mostrarán mediante el uso de tablas, que tendrán la siguiente plantilla:

[Periodo generado] / [Titulación del calendario generado]		
Nombre Restricción	Daño	Número de veces incumplida
Restricción 1	[Valor numérico impuesto al daño]	[Número de veces que no se cumple la restricción]
Restricción 2	[Valor numérico impuesto al daño]	[Número de veces que no se cumple la restricción]
Restricción 3	[Valor numérico impuesto al daño]	[Número de veces que no se cumple la restricción]
Restricción 4	[Valor numérico impuesto al daño]	[Número de veces que no se cumple la restricción]
Restricción 5	[Valor numérico impuesto al daño]	[Número de veces que no se cumple la restricción]
Restricción 6	[Valor numérico impuesto al daño]	[Número de veces que no se cumple la restricción]

Ilustración 28: Plantilla para las pruebas de la eficacia de los calendarios

El nombre de las restricciones hace referencia a cómo vienen definidas en el apartado referente al algoritmo 6.3 “Proceso del algoritmo utilizado”, para saber cuál es cuál dirigirse allí, donde se explica cada una.

Primer Cuatrimestre / Informática		
Nombre Restricción	Daño	Número de veces incumplida
Restricción 1	15	4
Restricción 2	25	1
Restricción 3	16	2
Restricción 4	10	8
Restricción 5	18	0
Restricción 6	15	3

Primer Cuatrimestre / Doble Grado Inf - ADE		
Nombre Restricción	Daño	Número de veces incumplida
Restricción 1	15	2
Restricción 2	25	0
Restricción 3	16	1
Restricción 4	10	4
Restricción 5	18	0
Restricción 6	15	3

Como podemos observar, los resultados son muy parecidos para los calendarios generados de las dos titulaciones. Los resultados son bastante buenos porque las restricciones que no deberían de infringirse nunca no lo hacen (excepto en Informática alguna vez), pero son mejorables también.

Podemos observar que la restricción número cuatro, la que hace referencia a que no pueda haber una asignatura el mismo día en distinto turno y del mismo curso y especialidad se incumple demasiadas veces, por lo que hay que mejorarlo.

La manera de mejorar esto es subiendo el daño que hacen las restricciones que se incumplen demasiado, pero al subirles el daño puede que se incumplan otras restricciones que sean más importantes, por lo que hay que tener cuidado. Como ya hemos comentado en la generación del calendario de la convocatoria extraordinaria se incumplirán muchas más veces las restricciones, pero es completamente normal ya que es imposible que no lo hicieran en una franja de días tan corta.

Aún así, tras haber realizado la primera prueba y haber consultado con el cliente nos hemos dado cuenta del orden que tienen que cumplir las distintas restricciones, de menor a mayor peso (por el número de la restricción): $5 < 1 < 4 < 6 < 3 < 2$.

La restricción cinco es la menos importante y la dos es la más importante.

Por tanto, haremos una segunda prueba subiendo el daño que tienen estas restricciones, de manera que el incumplimiento de las mismas sea menor. La primera vez se cogieron esos valores numéricos a “grosso modo”, como prueba inicial, pero esta vez, al estudiar los resultados obtenidos con esos valores se ha optado por llegar a un límite máximo de daño para una restricción de catorce, y los demás valores se han ido ajustando con respecto a eso, obteniendo los siguientes resultados:

Primer Cuatrimestre / Informática		
Nombre Restricción	Daño	Número de veces incumplida
Restricción 1	6	1
Restricción 2	14	0
Restricción 3	12	0
Restricción 4	8	2
Restricción 5	3	2
Restricción 6	2	4

Primer Cuatrimestre / Doble Grado Inf - ADE		
Nombre Restricción	Daño	Número de veces incumplida
Restricción 1	6	1
Restricción 2	14	0
Restricción 3	12	0
Restricción 4	8	0
Restricción 5	3	1
Restricción 6	2	4

Como podemos observar, los resultados obtenidos con estos valores son mucho mejores que los anteriores. Las restricciones que no deberían incumplirse nunca lo hacen, por tanto hemos notado que el cambio en los daños de las restricciones ha sido satisfactorio y creemos que los resultados obtenidos son óptimos, teniendo en cuenta la franja corta de doce días para tantas asignaturas, por lo que creemos que se han cumplido los objetivos correctamente. Podemos observar que la restricción sexta, que es la que hace referencia a la separación de las asignaturas difíciles se incumple varias veces, pero las dos razones de esto es el número de asignaturas difíciles que hemos establecido de entrada (que son siete), por lo que es difícil que estén separadas, y que le hemos asignado un daño menor a esa restricción, por una parte porque sus multiplicadores funcionan de distinta manera y por otra para asegurarnos de que las otras restricciones se cumplen mejor.

Como conclusión a estas pruebas podemos decir que los calendarios generados por nuestro algoritmo son óptimos, es lógico que alguna restricción se incumpla más de una vez porque hay muchos factores que debemos tener en cuenta, pero lo importante es que en un corto período de días se puedan cumplir las restricciones más importantes, y esto lo consigue de manera satisfactoria, además de cumplir siempre en el grado de lo posible la séptima restricción de la que aquí no hemos hablado, que es colocar los exámenes por la mañana siempre que se pueda.

8. Conclusiones y líneas futuras

En este apartado terminaremos exponiendo las conclusiones obtenidas en este proyecto, tanto a nivel técnico, como teórico y a nivel personal. También expondremos las líneas futuras que podría tener este proyecto, que son aquellas partes donde podría mejorarse para ofrecer una solución aún mejor, o poder encaminarse a conseguir otros objetivos más amplios para nuestro proyecto.

8.1 Conclusiones

En este Proyecto Fin de Carrera hemos desarrollado una solución para la generación automática de calendarios de exámenes de la Universidad, capaz de poder distinguir distintas restricciones y de cumplirlas, en el mayor grado posible. Esta solución cumple de forma satisfactoria con todos los requerimientos establecidos por el cliente, además varios de los requisitos establecidos inicialmente se han modificado para mejorar la usabilidad de la aplicación y hacerla más funcional.

Una de las cosas que he aprendido tras la realización de este proyecto ha sido la realización de todo el ciclo de vida de una solución software, ya que, aunque solo se trate de una simulación me he dado cuenta de lo importante que es definir y seguir un ciclo de vida de un proyecto para la correcta realización del mismo, que se refleja en la solución obtenida.

A nivel teórico he aprendido, aparte de esto, la constante investigación que se está llevando a cabo en el tema de la generación automática de horarios y del “Timetabling” en general, que lleva desarrollándose desde hace más de una década y aún está en constante desarrollo, investigando con multitud de diferentes algoritmos, haciendo combinaciones de estos algoritmos para mejorar los resultados, e inventando algoritmos nuevos para obtener mejores resultados cada vez.

También me he dado cuenta de que últimamente lo que se intenta es explorar aproximaciones que puedan funcionar con un nivel de generalidad más alto del que existe actualmente, sin importar que se trate de horarios de un centro de estudios, u horarios de una Universidad, o de una empresa para los empleados, ya que las investigaciones se suelen centrar en temas más concretos, porque buscar algo general es complicado si queremos obtener los mejores resultados para ese problema.

También he aprendido qué se usan distintos algoritmos que no sabía que podían utilizarse para estos fines, y también algoritmos que no conocía, como son los algoritmos genéticos, donde he visto ideas muy interesantes.

En cuanto al nivel técnico, he aprendido multitud de cosas gracias a la aplicación de escritorio que he desarrollado, como la programación con el framework de .NET y en concreto con: El lenguaje de programación C#, la programación orientada a objetos, la arquitectura de la aplicación, la interactividad de programación con la aplicación Excel, la gestión de hilos, el acceso a bases de datos y operaciones sobre ellas, y muchas cosas más. Aparte de la programación también he aprendido cómo montar una base de datos con SQL Server, cómo instalarla en otros sistemas y como realizar distintos tipos de consultas sobre ella para obtener conclusiones en mi aplicación.

Por último en cuanto a nivel personal, he aprendido a desarrollar una aplicación orientada a un cliente, aprendiendo distintas maneras de hacerla visualmente agradable, así como intuitiva, y con esto se aprende mucho para el futuro profesional.

Por tanto, pienso que el realizar una tarea como es un Proyecto Fin de Carrera, aparte de ser algo largo y costoso, también es una manera de obtener un conocimiento mayor acerca de lo que podemos llegar a hacer y de nuestras capacidades, y una aproximación a los futuros retos que nos surgirán a lo largo de nuestra carrera profesional. Por lo que creo que la realización de un proyecto de este tipo puede aportar mucho en todos los niveles a un estudiante que esté a punto de finalizar la carrera y facilitar la entrada al complicado mundo laboral al que más tarde o más temprano todos nos tendremos que enfrentar.

8.2 Líneas futuras

Como ya hemos visto nuestra aplicación cumple con los objetivos establecidos y tiene una buena eficiencia, pero aún así hay varias cosas que se pueden mejorar y ampliar, de manera que sea una aplicación con una funcionalidad mayor. La aplicación se ha hecho lo más abierta posible de acuerdo con los requisitos establecidos, dando a elegir al usuario cómo generar el calendario de exámenes, pero hay otras cosas que se podrían parametrizar para ofrecer al usuario más usabilidad.

Por tanto, se proponen varias líneas futuras para poder ampliar nuestra aplicación y mejorarla:

- Actualmente la aplicación genera los calendarios de exámenes asignándolos en el turno de mañana preferentemente, siempre que sea posible, antes que en el turno de tarde, por los requerimientos del cliente.
Se podría dar la opción al usuario de elegir si quiere que los exámenes se inserten preferentemente en el turno de mañana, en el turno de tarde, o que ninguno de los turnos tenga preferencia, de esta manera la aplicación mejoraría en usabilidad.
- Sería una buena idea también el parametrizar las distintas prioridades (daño) de las restricciones, de manera que el usuario pueda elegir qué restricción es más importante que otra. Esto puede ser útil por si en un futuro se decide que es más importante alguna restricción o que otra ha perdido importancia.
- También se podría implementar una funcionalidad que permitiera eliminar restricciones, por si el usuario quisiera que alguna restricción no se tuviera en cuenta (añadir sería imposible porque la lógica de cada restricción debe ir en el código).
- Desarrollar otros algoritmos para la generación del calendario de exámenes que puedan ser mejores o más óptimos, para poder investigar sobre los resultados y compararlos.

- Por último, y como línea futura más importante, se podría ampliar la aplicación para que permitiera la inclusión de más titulaciones, incluso de todas las titulaciones de la Universidad. Esto es una tarea más grande y que requeriría mucho más trabajo, puesto que tras realizar este proyecto se ha notado que solo estas dos titulaciones ya tienen sus particularidades, como las distintas especialidades; las asignaturas compartidas; las asignaturas “especiales”, que hay que tratar de distinta manera en el código (en este caso hay dos); etc. Además de tratar de manera distinta las titulaciones que comparten asignaturas con otras titulaciones (Dobles Grados).

Bibliografía

Aquí se exponen todos los recursos de información utilizados para la elaboración del proyecto, y diferenciamos las referencias correspondientes a libros y artículos consultados, y las referencias a páginas web de Internet.

Bibliografía

- (Burke, Newall & Weare, 1996). A memetic algorithm for University exam timetabling.
- (Burke, et. al., 2000). Structured cases in CBR – Re-using and adapting cases for time-tabling problems.
- (Burke & Petrovic, 2002). Recent research directions in automated timetabling.
- (Meisels, Kuflik & Gudes, 1991). Limited-resource scheduling by generalized rule-based system.

Recursos utilizados

- Alternativa para el proceso de planificación de horarios docentes de una Universidad.
<http://www.ciencias.holguin.cu/2007/Diciembre/articulos/ARTI6.htm>
- Qué es Linq.
<http://www.crazyteam.es/c/%C2%BFque-es-linq/>
- SQL Server Consola de comandos.
<http://www.docstoc.com/docs/31385582/SQL-Server---Consola-de-comandos-SQLCMD>
- Visual Studio 2010.
<http://msdn.microsoft.com/es-es/library/52f3sw5c>
- ¿Qué es UML?
<http://www.docirs.cl/uml.htm>
- Ciclo de vida del software.
<http://www.ia.uned.es/ia/asignaturas/adms/GuiaDidADMS/node10.html>
- Modelo Vista Controlador.
<http://blogdeaitor.wordpress.com/2008/10/20/model-view-controller/>
- Entity Framework.
<http://www.scip.be/index.php?Page=ArticlesNET12#EntityFramework>
- Excel y C Sharp.
<http://bcastilloblog.blogspot.com.es/2011/03/funciones-c-para-obtener-numero-de.html>

- UAC and C Sharp.
<http://victorhurdugaci.com/using-uac-with-c-part-1/>
- Background Worker in C Sharp.
<http://www.dotnetperls.com/backgroundworker>
- Linq to SQL.
<http://www.c-sharpcorner.com/uploadfile/scottlysle/linq-to-sql-in-C-Sharp/>
- Sistemas inteligentes – T3: Búsqueda Heurística. Universidad de Oviedo.
www.aic.uniovi.es/ssii/SSII-T3-BusquedaII.pdf

ANEXO A. Listado de Requisitos

En este anexo se expondrán todos los requisitos que tiene que cumplir la aplicación, tanto los requisitos de usuario como los requisitos de software.

Los requisitos de usuario son aquellos tratados con el cliente, es decir, todos los objetivos que tiene que cumplir nuestro sistema para que el cliente esté satisfecho con el producto.

Los requisitos de software son aquellos requisitos que son transparentes al usuario, pero necesarios para que la aplicación funcione correctamente. Describen el comportamiento del sistema a desarrollar, cómo deben actuar los distintos componentes de la aplicación.

El identificador del requisito consistirá en dos letras indicando el tipo de requisito seguidas de un guión y un número que será un contador.

A continuación se mostrará el listado formal de todos los requisitos con la siguiente plantilla:

[Identificador de requisito]	
Nombre	[Nombre del requisito]
Fuente	[Nombre de la fuente]
Prioridad	[Tipo de prioridad] (Alta, media, baja)
Necesidad	[Tipo de necesidad] (Esencial, deseable, opcional)
Claridad	[Tipo de claridad del requisito] (Alta, media, baja)
Estabilidad	[Tipo de estabilidad del requisito] (Alta, media, baja)
Descripción	[Descripción del requisito]
Verificabilidad	[Factor de verificabilidad del requisito] (Alta, media, baja)

Ilustración 29: Plantilla de requisito

A.1 Requisitos de usuario

RU-01			
Nombre	Fecha Inicio		
Fuente	Rubén Caballero Pérez <input checked="" type="checkbox"/>	Elena Castro Galán <input checked="" type="checkbox"/>	
Prioridad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>
Necesidad	Esencial <input checked="" type="checkbox"/>	Opcional <input type="checkbox"/>	
Claridad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>
Estabilidad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>
Descripción	El usuario elegirá la fecha de inicio que desee para que comience el calendario de exámenes.		
Verificabilidad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>

RU-02			
Nombre	Fecha fin		
Fuente	Rubén Caballero Pérez <input checked="" type="checkbox"/>	Elena Castro Galán <input checked="" type="checkbox"/>	
Prioridad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>
Necesidad	Esencial <input checked="" type="checkbox"/>	Opcional <input type="checkbox"/>	
Claridad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>
Estabilidad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>
Descripción	El usuario elegirá la fecha de fin que desee para que finalice el calendario de exámenes.		
Verificabilidad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>

RU-03			
Nombre	Curso		
Fuente	Rubén Caballero Pérez <input checked="" type="checkbox"/>	Elena Castro Galán <input checked="" type="checkbox"/>	
Prioridad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>
Necesidad	Esencial <input checked="" type="checkbox"/>		Opcional <input type="checkbox"/>
Claridad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>
Estabilidad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>
Descripción	Habrá un total de cuatro cursos para la titulación Grado en Ingeniería Informática, y un total de seis cursos para la titulación Doble Grado en Informática – ADE.		
Verificabilidad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>

RU-04			
Nombre	Cuatrimestre		
Fuente	Rubén Caballero Pérez <input checked="" type="checkbox"/>	Elena Castro Galán <input checked="" type="checkbox"/>	
Prioridad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>
Necesidad	Esencial <input checked="" type="checkbox"/>		Opcional <input type="checkbox"/>
Claridad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>
Estabilidad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>
Descripción	Habrá un total de dos cuatrimestres para las dos titulaciones, que serán los siguientes: Primer Cuatrimestre, Segundo Cuatrimestre.		
Verificabilidad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>

RU-05			
Nombre	Subtitulación		
Fuente	Rubén Caballero Pérez <input checked="" type="checkbox"/>	Elena Castro Galán <input checked="" type="checkbox"/>	
Prioridad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>
Necesidad	Esencial <input checked="" type="checkbox"/>		Opcional <input type="checkbox"/>
Claridad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>
Estabilidad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>
Descripción	<p>Cada asignatura tendrá un valor para subtitulación distinto dependiendo de la Titulación a la que pertenezca. Las asignaturas de Informática no tendrán subtitulación. Las asignaturas del Doble Grado podrán tener:</p> <ul style="list-style-type: none"> - Informática. - ADE. 		
Verificabilidad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>

RU-06			
Nombre	Especialidades		
Fuente	Rubén Caballero Pérez <input checked="" type="checkbox"/>	Elena Castro Galán <input checked="" type="checkbox"/>	
Prioridad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>
Necesidad	Esencial <input checked="" type="checkbox"/>		Opcional <input type="checkbox"/>
Claridad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>
Estabilidad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>
Descripción	<p>Todas las asignaturas tendrá una especialidad al menos, algunas pueden tener pertenecer a más de una especialidad. Las especialidades son las siguientes:</p> <ul style="list-style-type: none"> - Común. - Ingeniería de Computadores. - Sistemas de la Información. - Computación. 		
Verificabilidad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>

RU-07			
Nombre	Dificultad (Peso)		
Fuente	Rubén Caballero Pérez <input checked="" type="checkbox"/>	Elena Castro Galán <input checked="" type="checkbox"/>	
Prioridad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>
Necesidad	Esencial <input checked="" type="checkbox"/>		Opcional <input type="checkbox"/>
Claridad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>
Estabilidad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>
Descripción	Todas las asignaturas tendrán una dificultad. Esta dificultad será: - 1 = Normal. - 2 = Difícil.		
Verificabilidad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>

RU-08			
Nombre	Fecha Examen		
Fuente	Rubén Caballero Pérez <input checked="" type="checkbox"/>	Elena Castro Galán <input checked="" type="checkbox"/>	
Prioridad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>
Necesidad	Esencial <input checked="" type="checkbox"/>		Opcional <input type="checkbox"/>
Claridad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>
Estabilidad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>
Descripción	Todas las asignaturas deberán poder tener una fecha de examen. Inicialmente las asignaturas que no pertenezcan a la rama de ADE no tendrán fecha de examen.		
Verificabilidad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>

RU-09			
Nombre	Fecha Exámenes ADE		
Fuente	Rubén Caballero Pérez <input checked="" type="checkbox"/>	Elena Castro Galán <input checked="" type="checkbox"/>	
Prioridad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>
Necesidad	Esencial <input checked="" type="checkbox"/>		Opcional <input type="checkbox"/>
Claridad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>
Estabilidad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>
Descripción	Todas las asignaturas pertenecientes a la rama de ADE deberán tener obligatoriamente una fecha de examen, tanto inicialmente como finalmente, y deberá ser la misma.		
Verificabilidad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>

RU-10			
Nombre	Importar asignaturas		
Fuente	Rubén Caballero Pérez <input checked="" type="checkbox"/>	Elena Castro Galán <input checked="" type="checkbox"/>	
Prioridad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>
Necesidad	Esencial <input checked="" type="checkbox"/>		Opcional <input type="checkbox"/>
Claridad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>
Estabilidad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>
Descripción	El usuario podrá importar los datos de todas las asignaturas del año actual mediante un fichero de entrada que contendrá esos datos.		
Verificabilidad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>

RU-11			
Nombre	Elegir optativas		
Fuente	Rubén Caballero Pérez <input checked="" type="checkbox"/>	Elena Castro Galán <input checked="" type="checkbox"/>	
Prioridad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>
Necesidad	Esencial <input checked="" type="checkbox"/>		Opcional <input type="checkbox"/>
Claridad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>
Estabilidad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>
Descripción	El usuario podrá elegir las optativas que se vayan a impartir el año actual para la titulación del Grado en Informática.		
Verificabilidad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>

RU-12			
Nombre	Visualizar optativas		
Fuente	Rubén Caballero Pérez <input checked="" type="checkbox"/>	Elena Castro Galán <input type="checkbox"/>	
Prioridad	Alta <input type="checkbox"/>	Media <input checked="" type="checkbox"/>	Baja <input type="checkbox"/>
Necesidad	Esencial <input type="checkbox"/>		Opcional <input checked="" type="checkbox"/>
Claridad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>
Estabilidad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>
Descripción	El usuario podrá visualizar las asignaturas optativas que ha elegido para el año actual si lo desea.		
Verificabilidad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>

RU-13			
Nombre	Mostrar asignaturas		
Fuente	Rubén Caballero Pérez <input checked="" type="checkbox"/>	Elena Castro Galán <input type="checkbox"/>	
Prioridad	Alta <input type="checkbox"/>	Media <input checked="" type="checkbox"/>	Baja <input type="checkbox"/>
Necesidad	Esencial <input type="checkbox"/>		Opcional <input checked="" type="checkbox"/>
Claridad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>
Estabilidad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>
Descripción	El usuario podrá visualizar los datos de las asignaturas que hay en el año actual de cualquiera de las dos titulaciones si lo desea.		
Verificabilidad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>

RU-14			
Nombre	Elección del período		
Fuente	Rubén Caballero Pérez <input checked="" type="checkbox"/>	Elena Castro Galán <input checked="" type="checkbox"/>	
Prioridad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>
Necesidad	Esencial <input checked="" type="checkbox"/>		Opcional <input type="checkbox"/>
Claridad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>
Estabilidad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>
Descripción	<p>El usuario deberá elegir un período para realizar la generación del calendario de exámenes para ese período. Los posibles periodos a elegir son:</p> <ul style="list-style-type: none"> - Primer cuatrimestre. - Segundo cuatrimestre. - Convocatoria extraordinaria. 		
Verificabilidad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>

RU-15			
Nombre	Elección de sábados		
Fuente	Rubén Caballero Pérez <input checked="" type="checkbox"/>	Elena Castro Galán <input checked="" type="checkbox"/>	
Prioridad	Alta <input type="checkbox"/>	Media <input checked="" type="checkbox"/>	Baja <input type="checkbox"/>
Necesidad	Esencial <input checked="" type="checkbox"/>		Opcional <input type="checkbox"/>
Claridad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>
Estabilidad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>
Descripción	El usuario tendrá la opción de elegir si desea que los días que sean sábado se tengan en cuenta para la generación del calendario de exámenes.		
Verificabilidad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>

RU-16			
Nombre	Elección de algoritmo		
Fuente	Rubén Caballero Pérez <input checked="" type="checkbox"/>	Elena Castro Galán <input checked="" type="checkbox"/>	
Prioridad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>
Necesidad	Esencial <input checked="" type="checkbox"/>		Opcional <input type="checkbox"/>
Claridad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>
Estabilidad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>
Descripción	El usuario deberá elegir un algoritmo para la generación del calendario de exámenes.		
Verificabilidad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>

RU-17			
Nombre	Turnos		
Fuente	Rubén Caballero Pérez <input checked="" type="checkbox"/>	Elena Castro Galán <input checked="" type="checkbox"/>	
Prioridad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>
Necesidad	Esencial <input checked="" type="checkbox"/>		Opcional <input type="checkbox"/>
Claridad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>
Estabilidad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>
Descripción	<p>Cada día del calendario de exámenes tendrá dos turnos donde en cada turno podrá ir más de una asignatura (estarán solapadas). Los turnos son los siguientes:</p> <ul style="list-style-type: none"> - Turno de mañana. - Turno de tarde. 		
Verificabilidad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>

RU-18			
Nombre	Domingos		
Fuente	Rubén Caballero Pérez <input checked="" type="checkbox"/>	Elena Castro Galán <input checked="" type="checkbox"/>	
Prioridad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>
Necesidad	Esencial <input checked="" type="checkbox"/>		Opcional <input type="checkbox"/>
Claridad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>
Estabilidad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>
Descripción	<p>Los días que sean domingo no deberán tenerse en cuenta a la hora de generar el calendario de exámenes, ya que esos días no podrá haber ningún examen.</p>		
Verificabilidad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>

RU-19			
Nombre	Datos de entrada		
Fuente	Rubén Caballero Pérez <input checked="" type="checkbox"/>	Elena Castro Galán <input checked="" type="checkbox"/>	
Prioridad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>
Necesidad	Esencial <input checked="" type="checkbox"/>		Opcional <input type="checkbox"/>
Claridad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>
Estabilidad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>
Descripción	Los datos de entrada referentes a las asignaturas podrán ser modificados y leídos de forma fácil para el usuario, de manera que pueda ver de manera clara todos los datos de las asignaturas que va a introducir.		
Verificabilidad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>

RU-20			
Nombre	Datos de entrada no válidos		
Fuente	Rubén Caballero Pérez <input checked="" type="checkbox"/>	Elena Castro Galán <input checked="" type="checkbox"/>	
Prioridad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>
Necesidad	Esencial <input checked="" type="checkbox"/>		Opcional <input type="checkbox"/>
Claridad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>
Estabilidad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>
Descripción	Debe existir una manera de evitar que el usuario introduzca valores no válidos en los datos de entrada de las asignaturas. Deberá haber una validación para todos los datos.		
Verificabilidad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>

RU-21			
Nombre	Turno sábados		
Fuente	Rubén Caballero Pérez <input checked="" type="checkbox"/>	Elena Castro Galán <input checked="" type="checkbox"/>	
Prioridad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>
Necesidad	Esencial <input checked="" type="checkbox"/>		Opcional <input type="checkbox"/>
Claridad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>
Estabilidad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>
Descripción	Los días que sean sábado solo tendrán un turno para poder colocar asignaturas: el turno de mañana, ya que los sábados por la tarde no podrá haber exámenes de asignaturas.		
Verificabilidad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>

RU-22			
Nombre	Generación calendario		
Fuente	Rubén Caballero Pérez <input checked="" type="checkbox"/>	Elena Castro Galán <input checked="" type="checkbox"/>	
Prioridad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>
Necesidad	Esencial <input checked="" type="checkbox"/>		Opcional <input type="checkbox"/>
Claridad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>
Estabilidad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>
Descripción	Los exámenes de las asignaturas deberán colocarse en los huecos disponibles que tengamos en el período elegido, cumpliendo con las restricciones impuestas de la mejor manera.		
Verificabilidad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>

RU-23			
Nombre	Almacenamiento persistente de asignaturas		
Fuente	Rubén Caballero Pérez <input checked="" type="checkbox"/>	Elena Castro Galán <input checked="" type="checkbox"/>	
Prioridad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>
Necesidad	Esencial <input checked="" type="checkbox"/>		Opcional <input type="checkbox"/>
Claridad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>
Estabilidad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>
Descripción	Las asignaturas del año actual deberán guardarse de forma persistente en algún tipo de almacenamiento de datos, para que el usuario no tenga que importar siempre los datos de entrada cada vez que ejecuta la aplicación. Este tipo de almacenamiento tendrá que ser una base de datos.		
Verificabilidad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>

RU-24			
Nombre	Días adyacentes (Restricción algoritmo)		
Fuente	Rubén Caballero Pérez <input checked="" type="checkbox"/>	Elena Castro Galán <input checked="" type="checkbox"/>	
Prioridad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>
Necesidad	Esencial <input checked="" type="checkbox"/>		Opcional <input type="checkbox"/>
Claridad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>
Estabilidad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>
Descripción	En días adyacentes del calendario de exámenes no podrá haber exámenes de asignaturas que sean del mismo curso y especialidad.		
Verificabilidad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>

RU-25			
Nombre	Mismo día y turno, y mismo curso (Restricción algoritmo)		
Fuente	Rubén Caballero Pérez <input checked="" type="checkbox"/>	Elena Castro Galán <input checked="" type="checkbox"/>	
Prioridad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>
Necesidad	Esencial <input checked="" type="checkbox"/>		Opcional <input type="checkbox"/>
Claridad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>
Estabilidad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>
Descripción	No podrá haber exámenes de asignaturas que sean del mismo curso y misma especialidad el mismo día y turno del calendario.		
Verificabilidad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>

RU-26			
Nombre	Mismo día y turno, y curso consecutivo (Restricción algoritmo)		
Fuente	Rubén Caballero Pérez <input checked="" type="checkbox"/>	Elena Castro Galán <input checked="" type="checkbox"/>	
Prioridad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>
Necesidad	Esencial <input checked="" type="checkbox"/>		Opcional <input type="checkbox"/>
Claridad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>
Estabilidad	Alta <input type="checkbox"/>	Media <input checked="" type="checkbox"/>	Baja <input type="checkbox"/>
Descripción	No podrá haber exámenes de asignaturas que sean de cursos consecutivos y misma especialidad el mismo día y turno del calendario.		
Verificabilidad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>

RU-27			
Nombre	Mismo día y distinto turno, y mismo curso (Restricción algoritmo)		
Fuente	Rubén Caballero Pérez <input checked="" type="checkbox"/>	Elena Castro Galán <input checked="" type="checkbox"/>	
Prioridad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>
Necesidad	Esencial <input checked="" type="checkbox"/>		Opcional <input type="checkbox"/>
Claridad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>
Estabilidad	Alta <input type="checkbox"/>	Media <input checked="" type="checkbox"/>	Baja <input type="checkbox"/>
Descripción	No podrá haber exámenes de asignaturas que sean del mismo curso y especialidad el mismo día pero en distinto turno.		
Verificabilidad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>

RU-28			
Nombre	Mismo día y distinto turno, y cursos consecutivos (Restricción algoritmo)		
Fuente	Rubén Caballero Pérez <input checked="" type="checkbox"/>	Elena Castro Galán <input checked="" type="checkbox"/>	
Prioridad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>
Necesidad	Esencial <input checked="" type="checkbox"/>		Opcional <input type="checkbox"/>
Claridad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>
Estabilidad	Alta <input type="checkbox"/>	Media <input checked="" type="checkbox"/>	Baja <input type="checkbox"/>
Descripción	No podrá haber exámenes de asignaturas que sean de cursos consecutivos y misma especialidad el mismo día pero en distinto turno.		
Verificabilidad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>

RU-29			
Nombre	Tratamiento dificultad alta (Restricción algoritmo)		
Fuente	Rubén Caballero Pérez <input checked="" type="checkbox"/>	Elena Castro Galán <input checked="" type="checkbox"/>	
Prioridad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>
Necesidad	Esencial <input checked="" type="checkbox"/>		Opcional <input type="checkbox"/>
Claridad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>
Estabilidad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>
Descripción	Los exámenes pertenecientes a asignaturas del mismo curso que tengan dificultad alta deberán estar separados lo máximo posible unos de otros, de manera que a los alumnos no tengan un examen de una asignatura difícil cercano a otro examen de otra asignatura que también sea difícil.		
Verificabilidad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>

RU-30			
Nombre	Preferencia turno de mañana (Restricción algoritmo)		
Fuente	Rubén Caballero Pérez <input checked="" type="checkbox"/>	Elena Castro Galán <input checked="" type="checkbox"/>	
Prioridad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>
Necesidad	Esencial <input checked="" type="checkbox"/>		Opcional <input type="checkbox"/>
Claridad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>
Estabilidad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>
Descripción	Los exámenes de las asignaturas deberían estar colocados en los turnos de mañana, siempre que sea posible.		
Verificabilidad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>

RU-31			
Nombre	Almacenamiento del calendario generado		
Fuente	Rubén Caballero Pérez <input checked="" type="checkbox"/>	Elena Castro Galán <input checked="" type="checkbox"/>	
Prioridad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>
Necesidad	Esencial <input checked="" type="checkbox"/>		Opcional <input type="checkbox"/>
Claridad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>
Estabilidad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>
Descripción	La generación del calendario de exámenes deberá de guardarse de forma persistente en algún tipo de almacenamiento de datos.		
Verificabilidad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>

RU-32			
Nombre	Formato calendario generado		
Fuente	Rubén Caballero Pérez <input checked="" type="checkbox"/>	Elena Castro Galán <input checked="" type="checkbox"/>	
Prioridad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>
Necesidad	Esencial <input checked="" type="checkbox"/>		Opcional <input type="checkbox"/>
Claridad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>
Estabilidad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>
Descripción	El formato de salida del calendario de exámenes generado deberá ser elegante, simple y fácil de leer para el usuario, sin estar muy recargado, con los datos relevantes de cada asignatura.		
Verificabilidad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>

RU-33			
Nombre	Elección del calendario a generar		
Fuente	Rubén Caballero Pérez <input checked="" type="checkbox"/>	Elena Castro Galán <input checked="" type="checkbox"/>	
Prioridad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>
Necesidad	Esencial <input checked="" type="checkbox"/>		Opcional <input type="checkbox"/>
Claridad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>
Estabilidad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>
Descripción	El usuario podrá elegir para qué curso querrá exportar el calendario de exámenes, o para todos los cursos. De igual manera, podrá elegir para qué titulación quiere exportar el calendario de exámenes.		
Verificabilidad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>

A.2 Requisitos de software

RS-01			
Nombre	Inicio aplicación		
Fuente	Rubén Caballero Pérez <input checked="" type="checkbox"/>	Elena Castro Galán <input checked="" type="checkbox"/>	
Prioridad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>
Necesidad	Esencial <input checked="" type="checkbox"/>	Opcional <input type="checkbox"/>	
Claridad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>
Estabilidad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>
Descripción	Al iniciar la aplicación el usuario visualizará la pantalla inicial, donde podrá importar las asignaturas de entrada.		
Verificabilidad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>

RS-02			
Nombre	Avance por pestañas		
Fuente	Rubén Caballero Pérez <input checked="" type="checkbox"/>	Elena Castro Galán <input type="checkbox"/>	
Prioridad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>
Necesidad	Esencial <input checked="" type="checkbox"/>	Opcional <input type="checkbox"/>	
Claridad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>
Estabilidad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>
Descripción	El usuario irá avanzando de pestaña en pestaña por la pantalla principal de la aplicación, eligiendo las opciones que desee para la generación del calendario. Estas pestañas serán “pasos” que el usuario debe ir cumpliendo.		
Verificabilidad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>

RS-03			
Nombre	Botón “Avanzar”		
Fuente	Rubén Caballero Pérez <input checked="" type="checkbox"/>	Elena Castro Galán <input checked="" type="checkbox"/>	
Prioridad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>
Necesidad	Esencial <input checked="" type="checkbox"/>		Opcional <input type="checkbox"/>
Claridad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>
Estabilidad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>
Descripción	Habrá un botón que el usuario debe pinchar para avanzar al paso siguiente de la aplicación. Podrá pulsarlo en cualquier momento.		
Verificabilidad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>

RS-04			
Nombre	Botón “Retroceder”		
Fuente	Rubén Caballero Pérez <input checked="" type="checkbox"/>	Elena Castro Galán <input checked="" type="checkbox"/>	
Prioridad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>
Necesidad	Esencial <input checked="" type="checkbox"/>		Opcional <input type="checkbox"/>
Claridad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>
Estabilidad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>
Descripción	Habrá un botón que el usuario debe pinchar para retroceder al paso siguiente de la aplicación. Podrá pulsarlo en cualquier momento.		
Verificabilidad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>

RS-05			
Nombre	Botón “Salir”		
Fuente	Rubén Caballero Pérez <input checked="" type="checkbox"/>	Elena Castro Galán <input checked="" type="checkbox"/>	
Prioridad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>
Necesidad	Esencial <input checked="" type="checkbox"/>		Opcional <input type="checkbox"/>
Claridad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>
Estabilidad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>
Descripción	Habrá un botón que el usuario deberá pinchar salir completamente de la aplicación.		
Verificabilidad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>

RS-06			
Nombre	Botón “Importar asignaturas”		
Fuente	Rubén Caballero Pérez <input checked="" type="checkbox"/>	Elena Castro Galán <input checked="" type="checkbox"/>	
Prioridad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>
Necesidad	Esencial <input checked="" type="checkbox"/>		Opcional <input type="checkbox"/>
Claridad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>
Estabilidad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>
Descripción	Existirá un botón en el que si el usuario pincha podrá importar los datos de entrada de las asignaturas eligiendo un fichero Excel.		
Verificabilidad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>

RS-07			
Nombre	Elegir Excel		
Fuente	Rubén Caballero Pérez <input checked="" type="checkbox"/>	Elena Castro Galán <input checked="" type="checkbox"/>	
Prioridad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>
Necesidad	Esencial <input checked="" type="checkbox"/>		Opcional <input type="checkbox"/>
Claridad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>
Estabilidad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>
Descripción	El usuario tendrá la opción de elegir el fichero de entrada Excel que desee donde estén contenidos los datos de las asignaturas.		
Verificabilidad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>

RS-08			
Nombre	Botón “Elegir optativas”		
Fuente	Rubén Caballero Pérez <input checked="" type="checkbox"/>	Elena Castro Galán <input checked="" type="checkbox"/>	
Prioridad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>
Necesidad	Esencial <input checked="" type="checkbox"/>		Opcional <input type="checkbox"/>
Claridad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>
Estabilidad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>
Descripción	Habrá un botón en el que el usuario podrá pinchar para que elija las asignaturas optativas que se vayan a impartir en el año actual.		
Verificabilidad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>

RS-09			
Nombre	Botón “Mostrar optativas”		
Fuente	Rubén Caballero Pérez <input checked="" type="checkbox"/>	Elena Castro Galán <input checked="" type="checkbox"/>	
Prioridad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>
Necesidad	Esencial <input type="checkbox"/>		Opcional <input checked="" type="checkbox"/>
Claridad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>
Estabilidad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>
Descripción	Habrá un botón en el que el usuario podrá pinchar para ver qué asignaturas optativas ha elegido para el año actual.		
Verificabilidad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>

RS-10			
Nombre	Botón “Mostrar asignaturas Informática”		
Fuente	Rubén Caballero Pérez <input checked="" type="checkbox"/>	Elena Castro Galán <input checked="" type="checkbox"/>	
Prioridad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>
Necesidad	Esencial <input type="checkbox"/>		Opcional <input checked="" type="checkbox"/>
Claridad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>
Estabilidad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>
Descripción	Habrá un botón en el que el usuario podrá pinchar para ver las asignaturas pertenecientes a la titulación Grado en Ingeniería Informática que existen para el año actual.		
Verificabilidad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>

RS-11			
Nombre	Botón “Mostrar asignaturas Doble Grado”		
Fuente	Rubén Caballero Pérez <input checked="" type="checkbox"/>	Elena Castro Galán <input checked="" type="checkbox"/>	
Prioridad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>
Necesidad	Esencial <input type="checkbox"/>		Opcional <input checked="" type="checkbox"/>
Claridad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>
Estabilidad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>
Descripción	Habrá un botón en el que el usuario podrá pinchar para ver las asignaturas pertenecientes a la titulación Doble Grado en Informática – ADE que existen para el año actual.		
Verificabilidad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>

RS-12			
Nombre	Caja de selección del cuatrimestre		
Fuente	Rubén Caballero Pérez <input checked="" type="checkbox"/>	Elena Castro Galán <input checked="" type="checkbox"/>	
Prioridad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>
Necesidad	Esencial <input checked="" type="checkbox"/>		Opcional <input type="checkbox"/>
Claridad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>
Estabilidad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>
Descripción	Se mostrarán tres Radio Button para que el usuario elija el periodo a generar para el calendario de exámenes.		
Verificabilidad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>

RS-13			
Nombre	Controles de fechas		
Fuente	Rubén Caballero Pérez <input checked="" type="checkbox"/>	Elena Castro Galán <input checked="" type="checkbox"/>	
Prioridad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>
Necesidad	Esencial <input checked="" type="checkbox"/>		Opcional <input type="checkbox"/>
Claridad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>
Estabilidad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>
Descripción	El usuario podrá elegir las fechas de inicio y de fin del calendario de exámenes mediante cajas de selección de fechas.		
Verificabilidad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>

RS-14			
Nombre	Caja de selección de sábados		
Fuente	Rubén Caballero Pérez <input checked="" type="checkbox"/>	Elena Castro Galán <input checked="" type="checkbox"/>	
Prioridad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>
Necesidad	Esencial <input checked="" type="checkbox"/>		Opcional <input type="checkbox"/>
Claridad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>
Estabilidad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>
Descripción	El usuario podrá elegir si quiere que se cuenten los sábados para el calendario mediante una caja de selección, que podrá marcar o dejar desmarcada.		
Verificabilidad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>

RS-15			
Nombre	Caja de selección de algoritmo		
Fuente	Rubén Caballero Pérez <input checked="" type="checkbox"/>	Elena Castro Galán <input checked="" type="checkbox"/>	
Prioridad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>
Necesidad	Esencial <input checked="" type="checkbox"/>		Opcional <input type="checkbox"/>
Claridad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>
Estabilidad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>
Descripción	El usuario podrá elegir el algoritmo encargado de generar el calendario de exámenes mediante un Radio Button.		
Verificabilidad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>

RS-16			
Nombre	Botón “Cancelar proceso”		
Fuente	Rubén Caballero Pérez <input checked="" type="checkbox"/>	Elena Castro Galán <input checked="" type="checkbox"/>	
Prioridad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>
Necesidad	Esencial <input type="checkbox"/>		Opcional <input checked="" type="checkbox"/>
Claridad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>
Estabilidad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>
Descripción	Deberá existir un botón en la aplicación que sea visible cuando el proceso de generación del algoritmo esté teniendo lugar, para que el usuario cancele el proceso si lo desea.		
Verificabilidad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>

RS-17			
Nombre	Botón “Acerca de”		
Fuente	Rubén Caballero Pérez <input checked="" type="checkbox"/>	Elena Castro Galán <input checked="" type="checkbox"/>	
Prioridad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>
Necesidad	Esencial <input type="checkbox"/>		Opcional <input checked="" type="checkbox"/>
Claridad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>
Estabilidad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>
Descripción	Existirá un botón en la aplicación en el que si el usuario pincha se visualizará información relativa a la versión de la aplicación, el autor de la misma y la fecha de salida.		
Verificabilidad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>

RS-18			
Nombre	Barra de progreso		
Fuente	Rubén Caballero Pérez <input checked="" type="checkbox"/>	Elena Castro Galán <input checked="" type="checkbox"/>	
Prioridad	Alta <input type="checkbox"/>	Media <input checked="" type="checkbox"/>	Baja <input type="checkbox"/>
Necesidad	Esencial <input type="checkbox"/>		Opcional <input checked="" type="checkbox"/>
Claridad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>
Estabilidad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>
Descripción	En el proceso de generación del calendario de exámenes el usuario podrá visualizar una barra de progreso que muestre además el porcentaje del proceso, a modo de comprobación.		
Verificabilidad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>

RS-19			
Nombre	Elección salida Titulación		
Fuente	Rubén Caballero Pérez <input checked="" type="checkbox"/>	Elena Castro Galán <input checked="" type="checkbox"/>	
Prioridad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>
Necesidad	Esencial <input checked="" type="checkbox"/>		Opcional <input type="checkbox"/>
Claridad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>
Estabilidad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>
Descripción	El usuario podrá elegir la Titulación que desee para exportar el calendario de exámenes mediante un Radio Button.		
Verificabilidad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>

RS-20			
Nombre	Elección salida Curso		
Fuente	Rubén Caballero Pérez <input checked="" type="checkbox"/>	Elena Castro Galán <input checked="" type="checkbox"/>	
Prioridad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>
Necesidad	Esencial <input checked="" type="checkbox"/>		Opcional <input type="checkbox"/>
Claridad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>
Estabilidad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>
Descripción	El usuario podrá elegir el curso que desee para exportar el calendario de exámenes mediante una lista desplegable donde se mostrarán todos los cursos.		
Verificabilidad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>

RS-21			
Nombre	Formato salida Calendario		
Fuente	Rubén Caballero Pérez <input checked="" type="checkbox"/>	Elena Castro Galán <input checked="" type="checkbox"/>	
Prioridad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>
Necesidad	Esencial <input checked="" type="checkbox"/>		Opcional <input type="checkbox"/>
Claridad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>
Estabilidad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>
Descripción	El calendario de exámenes generado se exportará a un fichero Excel por razones de claridad y elegancia.		
Verificabilidad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>

RS-22			
Nombre	Ejecutar proceso generador del calendario		
Fuente	Rubén Caballero Pérez <input checked="" type="checkbox"/>	Elena Castro Galán <input checked="" type="checkbox"/>	
Prioridad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>
Necesidad	Esencial <input checked="" type="checkbox"/>		Opcional <input type="checkbox"/>
Claridad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>
Estabilidad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>
Descripción	El usuario podrá ejecutar la generación del calendario de exámenes pulsando en el botón Generar, al encontrarse en el tercer paso.		
Verificabilidad	Alta <input checked="" type="checkbox"/>	Media <input type="checkbox"/>	Baja <input type="checkbox"/>

ANEXO B. Listado de Casos de Uso

En el presente anexo se describirá el listado completo de los distintos casos de uso de la aplicación. Ya hemos definido en el apartado 3.3 “Casos de Uso” lo que es un caso de uso, por lo que no nos centraremos en esto.

Tan solo es necesario recordar que un caso de uso es una acción que un actor puede realizar sobre el sistema, y que el actor podrá ser un usuario humano u otro sistema, aunque en nuestro caso, la acción siempre es realizada por un usuario humano.

El identificador del caso de uso serán las letras CU seguidas de un guión y un número de dos dígitos que será un contador.

A continuación se mostrará la plantilla para definir cada caso de uso de forma completa:

[Identificador de caso de uso]	
Nombre	[Nombre del caso de uso]
Actores	[Actor o actores del caso de uso]
Objetivo	[Objetivo del caso de uso]
Precondiciones	[Precondiciones del caso de uso]
Postcondiciones	[Postcondiciones del caso de uso]
Escenario básico	[Escenario típico del caso de uso]
Escenario alternativo	[Escenario alternativo del caso de uso]

Ilustración 30: Plantilla de caso de uso

CU-01	
Nombre	Avanzar paso
Actores	Usuario
Objetivo	Avanzar al paso siguiente en la aplicación
Precondiciones	Encontrarse en la pantalla principal. No estar en el último paso. Cumplir condiciones del paso actual.
Postcondiciones	Se avanzará al paso siguiente de la aplicación
Escenario básico	Pulsar en el botón “Avanzar”
Escenario alternativo	Ninguno

CU-02	
Nombre	Retroceder paso
Actores	Usuario
Objetivo	Retroceder al paso anterior en la aplicación
Precondiciones	Encontrarse en la pantalla principal. No estar en el primer paso.
Postcondiciones	Se retrocederá al paso anterior de la aplicación
Escenario básico	Pulsar en el botón “Retroceder”
Escenario alternativo	Ninguno

CU-03	
Nombre	Salir
Actores	Usuario
Objetivo	Salir completamente de la aplicación
Precondiciones	Encontrarse en la pantalla principal. No estar ejecutando proceso de generación del calendario.
Postcondiciones	Se cerrará completamente la aplicación
Escenario básico	Pulsar en el botón “Salir”
Escenario alternativo	Ninguno

CU-04	
Nombre	Importar asignaturas
Actores	Usuario
Objetivo	Importar las asignaturas del fichero de entrada a la base de datos
Precondiciones	Encontrarse en el paso 1.
Postcondiciones	Se elegirá el fichero de entrada a cargar. Se cargaran los datos de las asignaturas a la base de datos
Escenario básico	Pulsar en el botón “Importar asignaturas”
Escenario alternativo	Cuando el año actual no está en la base de datos, se inhabilitarán todos los controles excepto el botón de Importar.

CU-05	
Nombre	Mostrar asignaturas Informática
Actores	Usuario
Objetivo	Visualizar todas las asignaturas pertenecientes a la titulación Grado en Ingeniería Informática
Precondiciones	Encontrarse en el paso 1. Tener datos de asignaturas en la base de datos
Postcondiciones	Se mostrarán todas las asignaturas del Grado con todos sus datos
Escenario básico	Pulsar en el botón “Mostrar asignaturas Informática”
Escenario alternativo	Ninguno

CU-06	
Nombre	Mostrar asignaturas Doble Grado
Actores	Usuario
Objetivo	Visualizar todas las asignaturas pertenecientes a la titulación Doble Grado en Informática – ADE
Precondiciones	Encontrarse en el paso 1. Tener datos de asignaturas en la base de datos
Postcondiciones	Se mostrarán todas las asignaturas del Doble Grado con todos sus datos
Escenario básico	Pulsar en el botón “Mostrar asignaturas Doble Grado”
Escenario alternativo	Ninguno

CU-07	
Nombre	Elegir optativas
Actores	Usuario
Objetivo	Poder elegir las asignaturas optativas que se vayan a impartir el año actual pertenecientes a la titulación Grado en Ingeniería Informática
Precondiciones	Encontrarse en el paso 1. Tener datos de asignaturas en la base de datos
Postcondiciones	Se mostrarán todas las asignaturas optativas del Grado, y el usuario podrá elegir las que desee, o todas
Escenario básico	Pulsar en el botón “Elegir optativas Informática”
Escenario alternativo	Ninguno

CU-08	
Nombre	Visualizar optativas
Actores	Usuario
Objetivo	Mostrar las asignaturas optativas que el usuario ha elegido para el curso actual, a modo de consulta
Precondiciones	Encontrarse en el paso 1. Tener datos de asignaturas en la base de datos
Postcondiciones	Se mostrarán las asignaturas optativas que ha elegido el usuario. Si no ha elegido ninguna se mostrarán todas
Escenario básico	Pulsar en el botón “Mostrar optativas seleccionadas”
Escenario alternativo	Ninguno

CU-09	
Nombre	Elegir fechas
Actores	Usuario
Objetivo	Elegir el período deseado para generar el calendario y sus fechas
Precondiciones	Encontrarse en el paso 2
Postcondiciones	Se elegirán las fechas de inicio y de fin, así como el período que el usuario quiera para la generación
Escenario básico	Pulsar la opción del período deseado. Rellenar fecha de inicio y fecha de fin
Escenario alternativo	Ninguno

CU-10	
Nombre	Elegir algoritmo
Actores	Usuario
Objetivo	Elegir el algoritmo encargado de generar el calendario
Precondiciones	Encontrarse en el paso 3
Postcondiciones	Se elegirá el algoritmo deseado para la generación del calendario de exámenes
Escenario básico	Pulsar la opción deseada del algoritmo
Escenario alternativo	Ninguno

CU-11	
Nombre	Generar calendario
Actores	Usuario
Objetivo	Generar el calendario de exámenes deseado
Precondiciones	Encontrarse en el paso 3. Haber elegido un algoritmo.
Postcondiciones	Se ejecutará el proceso de generación. Luego se visualizarán los resultados por pantalla
Escenario básico	Pulsar el botón “Generar”
Escenario alternativo	Ninguno

CU-12	
Nombre	Cancelar generación
Actores	Usuario
Objetivo	Cancelar el proceso de generación del calendario
Precondiciones	Encontrarse en el paso 3. Estar ejecutándose la generación
Postcondiciones	Se cancelará el proceso de generación del calendario
Escenario básico	Pulsar el botón “Cancelar” (Mano con fondo rojo)
Escenario alternativo	Ninguno

CU-13	
Nombre	Exportar calendario
Actores	Usuario
Objetivo	Exportar el calendario generado anteriormente a un fichero Excel, para almacenarlo de forma persistente
Precondiciones	Encontrarse en el paso 4
Postcondiciones	Se exportará el calendario de exámenes a un fichero Excel con formato
Escenario básico	Elegir titulación. Elegir curso. Pinchar en el botón “Exportar”
Escenario alternativo	Ninguno

ANEXO C. Manual de instalación

En este anexo se presentará un completo manual de instalación para poder ejecutar correctamente la aplicación.

Los requisitos que necesitamos los podremos encontrar en el apartado 3.5 “Recursos para la construcción y funcionamiento del sistema”, aquí los repetiremos y explicaremos cómo instalar cada uno de los componentes necesarios para poder ejecutar la aplicación de forma correcta.

Es necesario tener instalado:

- .NET Framework 4.0.
- SQL Server 2008 Express.
- Instalación de la base de datos en nuestro servidor de SQL Express.
- Microsoft Office Excel.

La instalación de Microsoft Office Excel no la expondremos aquí, porque sabemos que no es un programa de distribución libre, y que los ordenadores de la Universidad ya lo tienen instalado, ya que dispone de las licencias para ello y viene en el pack de Microsoft Office.

A continuación separaremos por apartados la instalación de cada uno de los componentes necesarios para la aplicación.

Archivos adjuntos de la aplicación entregada para la correcta instalación

Con la solución software se incluyen los siguientes archivos, que son necesarios para la correcta instalación de la aplicación. Los archivos son los siguientes:

- Archivo de script SQL necesario para instalar la base de datos.
- Archivo ejecutable de script de terminal necesario para instalar el script SQL en la base de datos de forma rápida.

C.1 Instalación de .NET Framework 4

A continuación mostraremos cómo instalar el .NET Framework 4 de Microsoft. Debemos abrir un navegador web e ir a la siguiente dirección:

<http://www.microsoft.com/es-es/download/details.aspx?id=17851>

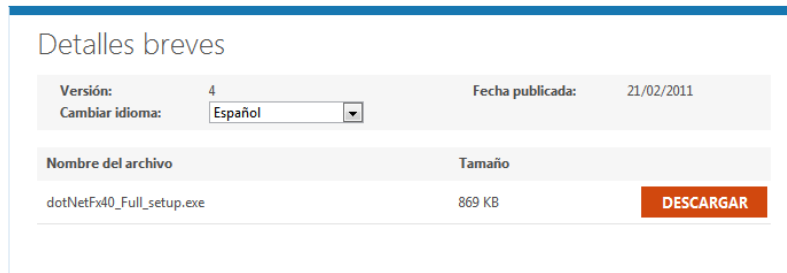


Ilustración 31: Página de descarga de .NET Framework 4

Una vez estemos en la página pulsaremos en el botón “Descargar”, como se muestra más arriba. Se nos descargará un archivo llamado “dotNetFx40_Full_setup.exe” que deberemos ejecutar.

Así pues, cuando ejecutemos este archivo comenzará la instalación, como vemos más abajo:

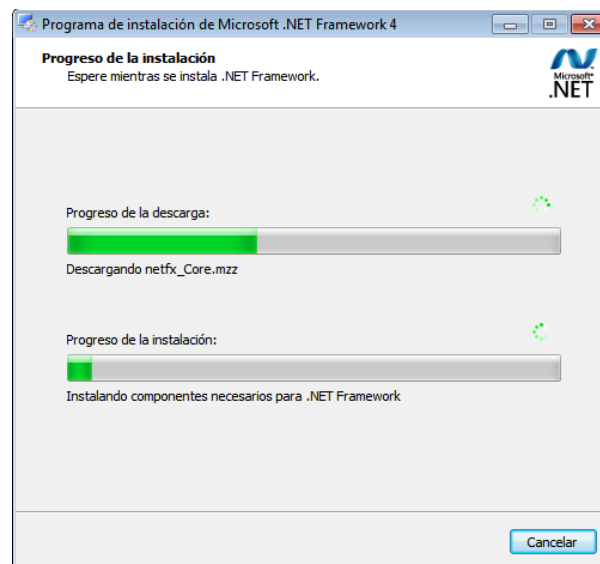


Ilustración 32: Instalando NET Framework 4

Una vez finalice la instalación de .NET Framework 4, si el sistema operativo nos pide que reiniciemos el equipo deberemos reiniciar, sino podremos continuar con la instalación de SQL Server 2008 Express.

C.2 Instalación de SQL Server 2008 Express

En este apartado mostraremos cómo instalar el Sistema Gestor de Base de Datos SQL Server 2008 Express Edition. Este software es de libre distribución y es gratuito, por lo que no habrá problemas en cuanto a la instalación.

Esta instalación es algo más larga que la instalación del .NET Framework 4, pero también es bastante fácil de instalar, iremos paso por paso. Para empezar, habrá que ir a la página de descarga con un navegador, como anteriormente. La dirección a la que tendremos que ir es:

<http://www.microsoft.com/es-es/download/details.aspx?id=1695>

En esta página tendremos tres opciones de descarga. Tenemos una llamada “SQLEXPRESS_x64_ESN.exe”, que es exclusivamente para sistemas operativos con una arquitectura de 64 bits; también tenemos otro archivo llamado “SQLEXPRESS_x86_ESN.exe”, el cual es compatible con sistemas operativos tanto de 32 bits como de 64 bits; por último tenemos el “SQLEXPRESS32_x86_ESN.exe”, el cual es una versión reducida exclusivamente para sistemas operativos de 32 bits.

Aconsejamos que si no se está seguro qué arquitectura tiene el sistema operativo instalado nos descarguemos el archivo llamado “SQLEXPRESS_x86_ESN.exe”, que sirve para las dos arquitecturas. Tendremos que pinchar en el botón descargar y luego bajarnos el archivo o ejecutarlo directamente.

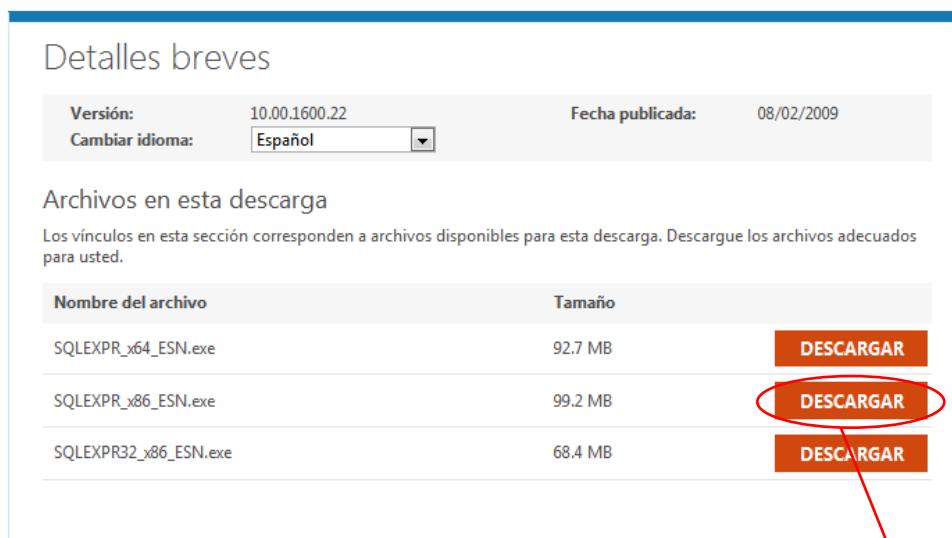


Ilustración 33: Descarga de SQL Server 2008 Express

Válido para las dos arquitecturas

Una vez se haya descargado el archivo correspondiente lo ejecutaremos. Nos aparecerá una ventana con varias opciones, como la siguiente:

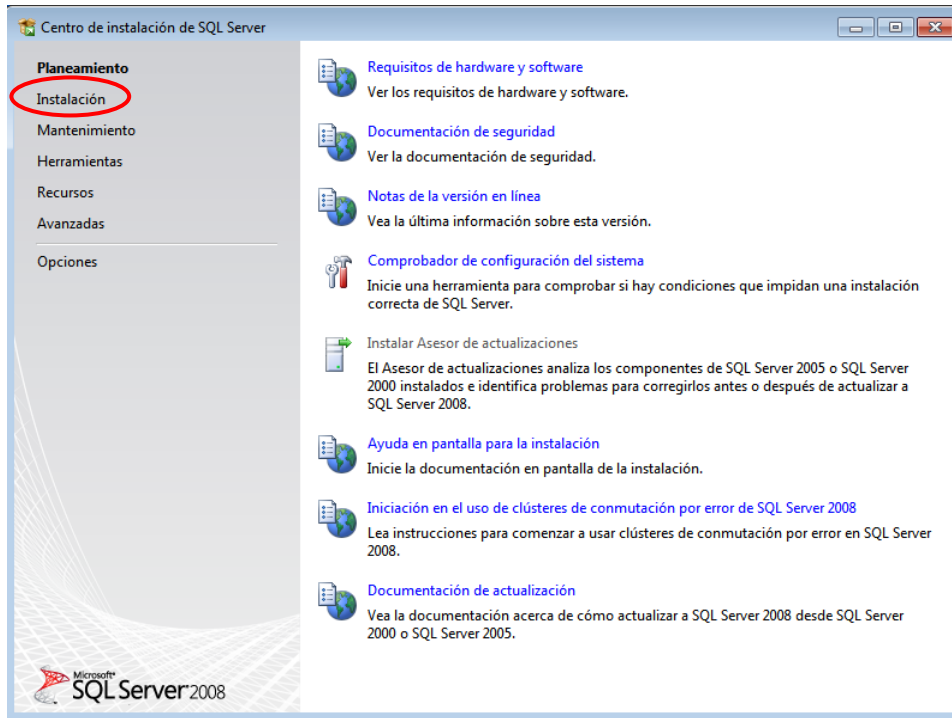


Ilustración 34: Instalación de SQL Server 2008 Express (1)

En esta ventana, en la parte izquierda, pincharemos en “Instalación”, como se muestra más arriba. A continuación nos mostrará otras opciones en la parte derecha donde deberemos pinchar en la primera, llamada “Nueva instalación independiente de SQL Server o agregar características a una instalación existente”.

Cuando hayamos seleccionado esa opción nos abrirá una nueva ventana donde comprobará una serie de archivos necesarios para la instalación. Pinchamos en el botón de abajo, que dice “Instalación”.

Nos aparecerá otra ventana donde comprobaremos una serie de reglas auxiliares que hay que cumplir para la instalación:

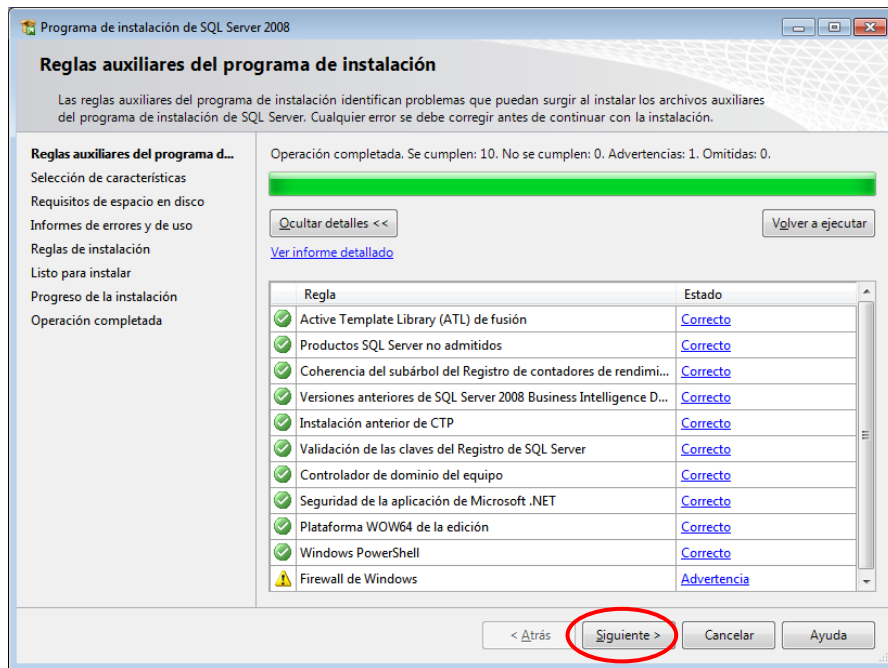


Ilustración 35: Instalación de SQL Server 2008 Express (2)

Pincharemos en el botón “Siguiente”. Nos aparecerá otra ventana con las características que queremos añadir a la instalación. Seleccionamos todas (hay un total de tres), y pulsaremos en el botón “Siguiente”.

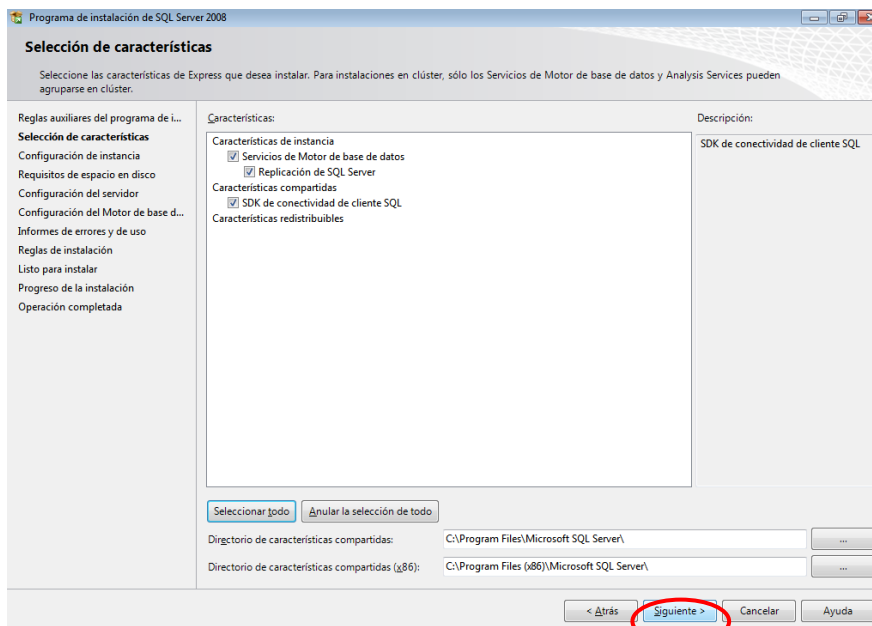


Ilustración 36: Instalación de SQL Server 2008 Express (3)

A continuación nos aparecerá una ventana donde podremos cambiar la configuración de la instancia de la base de datos, en esta ventana tendremos que dejar todo como está, sin cambiar nada, y pulsar en el botón “Siguiente”, como se muestra a continuación:

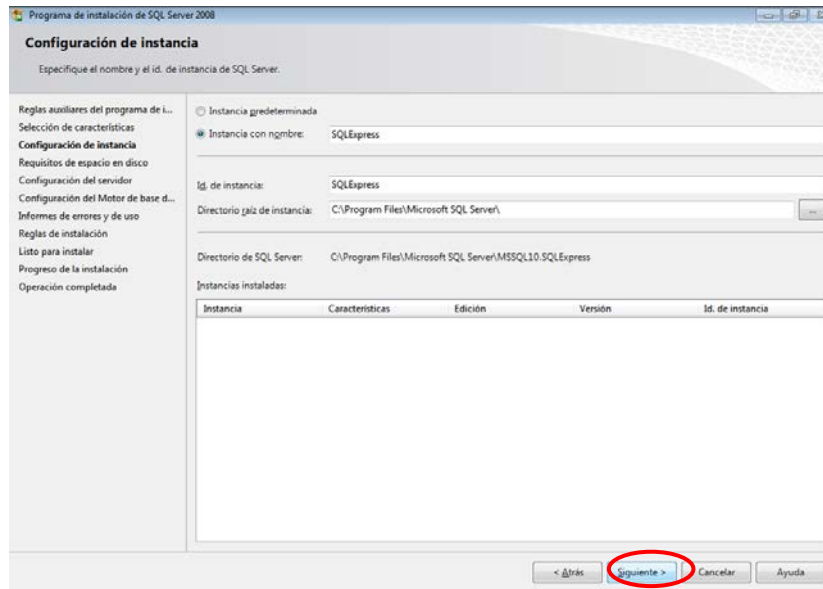


Ilustración 37: Instalación de SQL Server 2008 Express (4)

Nos aparecerá otra ventana donde nos mostrará el espacio de disco que ocupará la instalación, aquí simplemente debemos pulsar en el botón “Siguiente” de nuevo, para avanzar.

En la ventana siguiente nos mostrará la configuración del servidor. Aquí simplemente tendremos que seleccionar un nombre de cuenta para los servicios de la base de datos. Habrá que ponerlo como la ventana que mostramos a continuación y pulsar en el botón “Siguiente”:

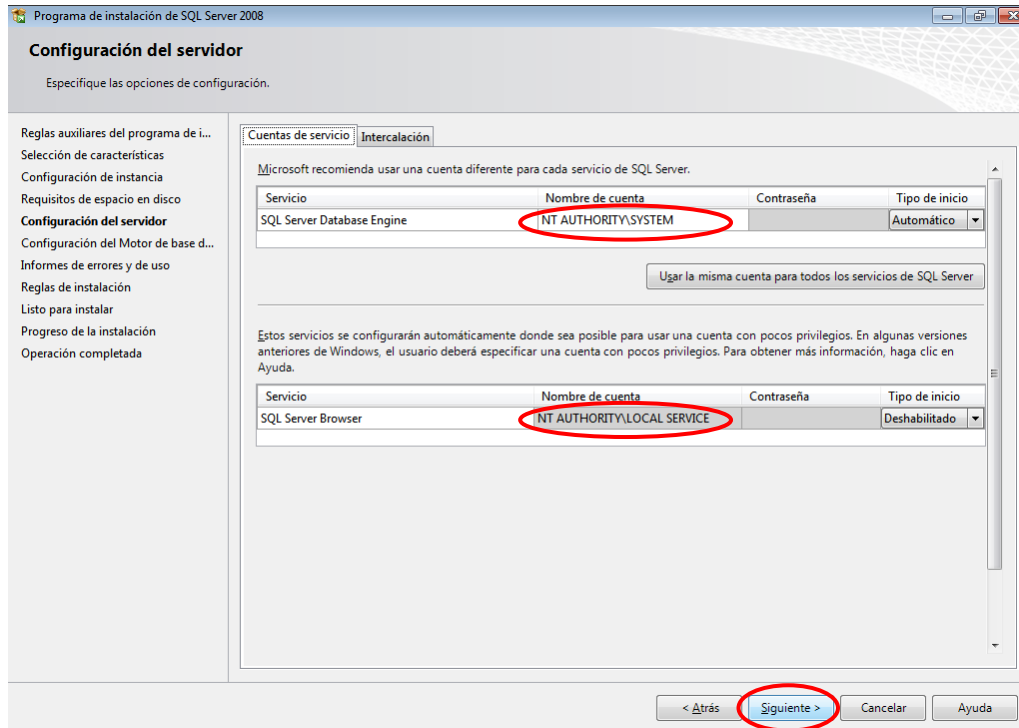


Ilustración 38: Instalación de SQL Server 2008 Express (5)

En el paso siguiente, nos aparecerá otra ventana con distintos aspectos de configuración del motor de la base de datos. Aquí lo único que deberemos hacer es especificar un administrador para el motor de la base de datos. Pincharemos en el botón “Agregar usuario actual”, y nos aparecerá nuestro usuario en la caja de abajo (en la imagen se ha quitado para que no se vea el código de usuario). A continuación pulsaremos en el botón “Siguiente”, para avanzar al siguiente paso de la instalación.

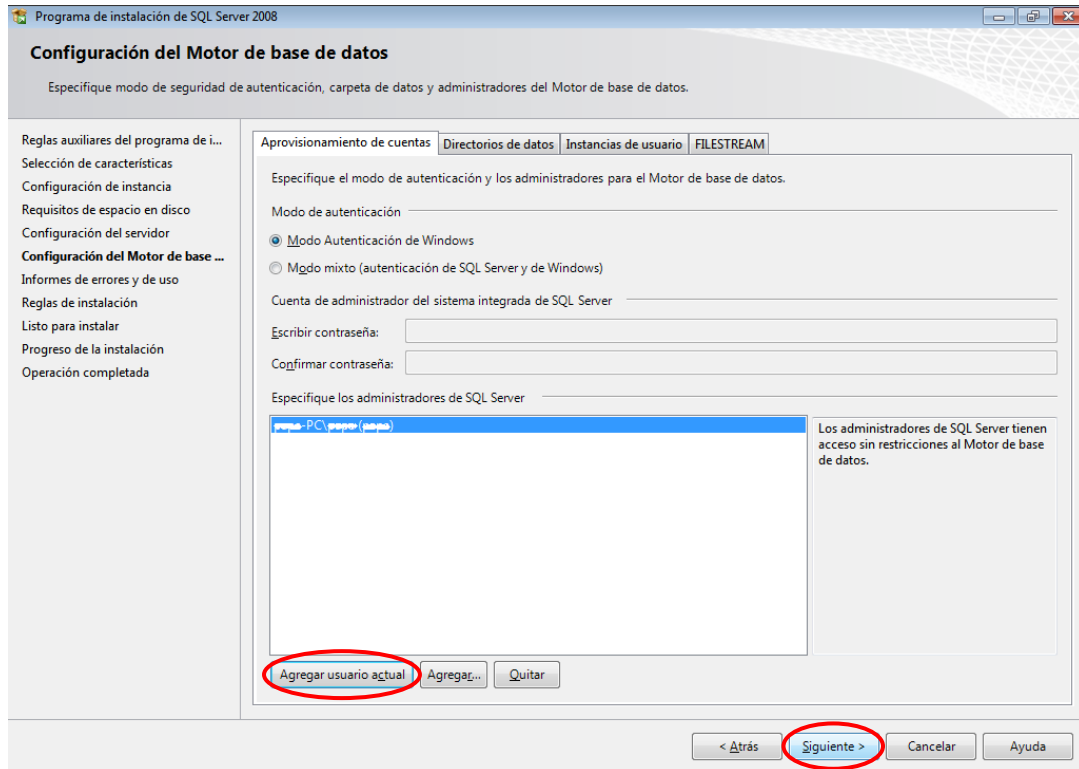


Ilustración 39: Instalación de SQL Server 2008 Express (6)

En la siguiente ventana nos preguntará si queremos enviar a Microsoft datos de uso e informes de errores. Esto podremos dejarlo como queramos, o dejarlo como está. Pulsamos en el botón “Siguiente”.

A continuación nos mostrará otra ventana donde el programa de instalación ejecutará unas reglas para ver si puede continuar con la instalación. Esperamos a que termine, y pulsamos de nuevo en el botón “Siguiente”.

Ahora nos mostrará todas las características y opciones que hemos seleccionado para instalar, solo tendremos que pulsar en el botón “Instalación”.

Por fin, ya se está instalando nuestro Sistema Gestor de Base de Datos. Esperamos a que se complete la instalación, y pulsaremos en el botón “Siguiente”:

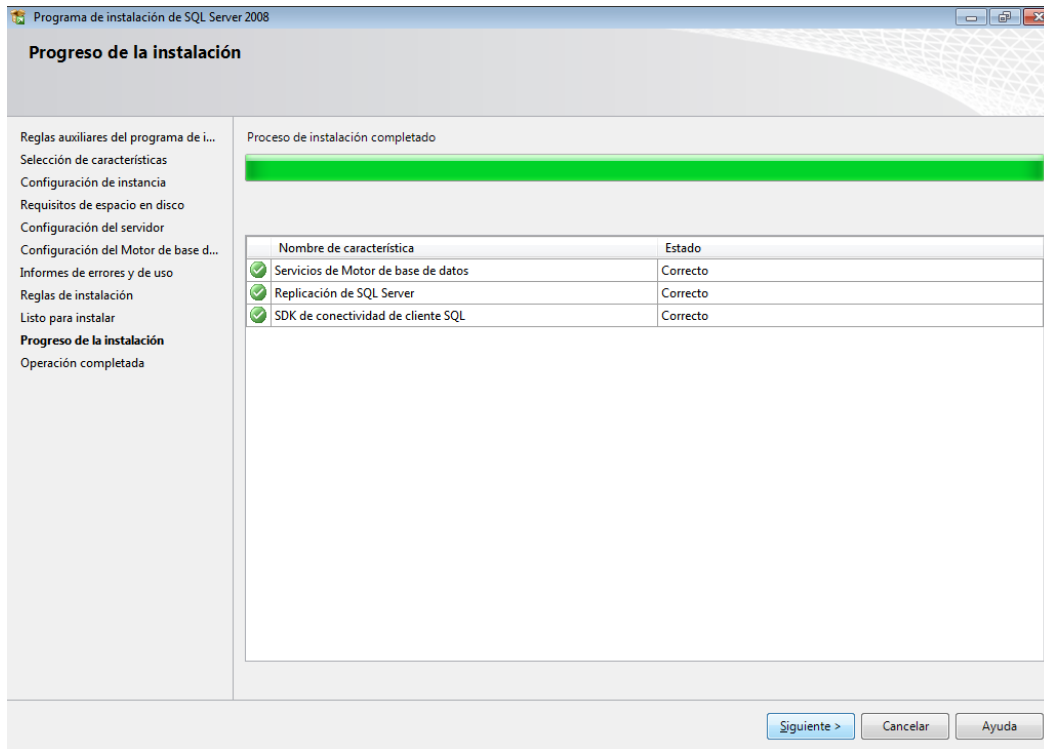


Ilustración 40: Instalación de SQL Server 2008 Express (7)

Y llegamos a la última pantalla de instalación, donde nos dice que la instalación ha sido completada, y que ha guardado un registro de resumen de la instalación, que podremos consultar si queremos, aunque si no ha habido ningún error no será necesario. Por tanto, pulsamos en el botón “Cerrar” y listo, habremos instalado SQL Server 2008 Express en nuestro ordenador.

C.3 Instalación de la base de datos en SQL Server

En este apartado explicaremos cómo instalar la base de datos inicial en nuestro SQL Server, para que la aplicación funcione de forma correcta. Esto es necesario, porque la aplicación no crea y define las tablas, pero sí las utiliza.

Por tanto, necesitaremos primero instalar nuestra base de datos en nuestro Sistema Gestor de Base de Datos.

Para esta labor, se ha hecho un sencillo script de instalación, el cual ejecuta otro script en lenguaje Transact-SQL, para que lo ejecute en nuestra instancia de SQL Server. Se ha hecho esto para facilitar esta labor al usuario y que simplemente tenga que ejecutar nuestro archivo de script.

Así pues, el usuario simplemente tendrá que abrir la carpeta llamada “ScriptInstalacion” y ejecutar el archivo “InstalacionScript.bat”.

Al ejecutar nuestro pequeño script nos aparecerá la siguiente ventana, indicándonos que se procederá a instalar la base de datos “CalendarioExamenes” en nuestro servidor, y que si queremos continuar. Debemos pulsar la tecla “s”:

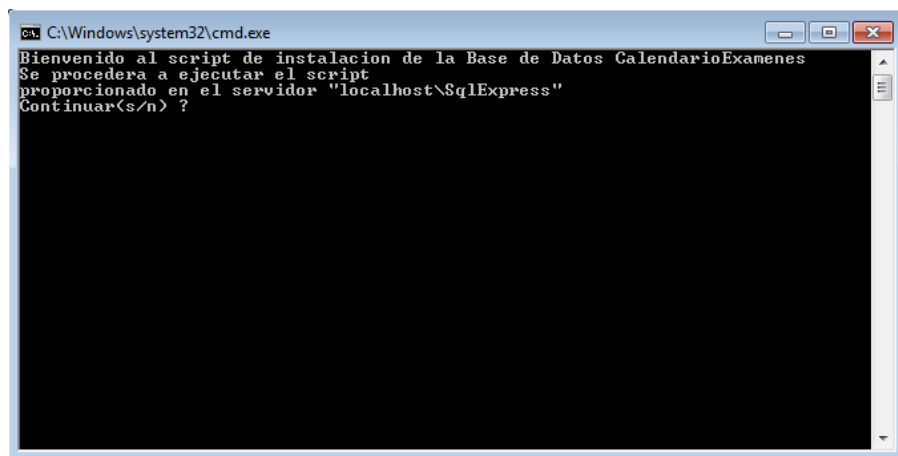
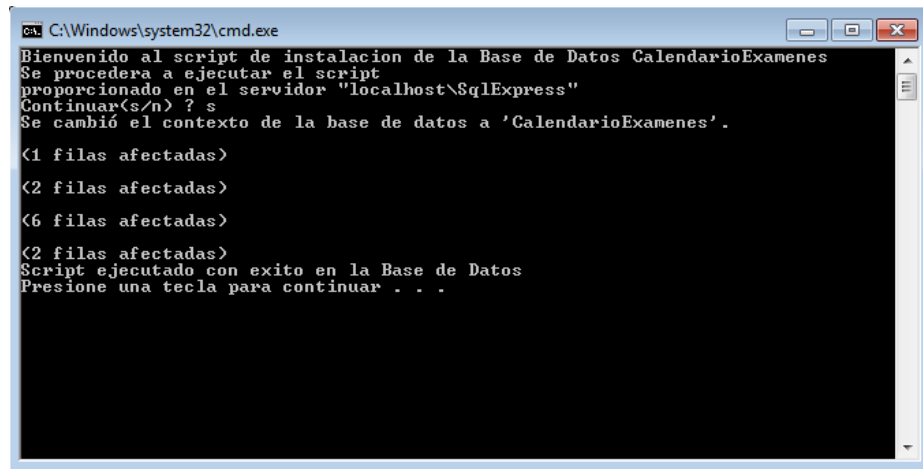


Ilustración 41: Instalación de la base de datos

Al pulsar la tecla “s” se ejecutará el script de instalación SQL en la base de datos y nos dirá que lo ha ejecutado con éxito, como se muestra en la pantalla siguiente:

A screenshot of a Windows command prompt window titled 'C:\Windows\system32\cmd.exe'. The window has a black background with white text. The text inside the window reads: 'Bienvenido al script de instalacion de la Base de Datos CalendarioExamenes', 'Se procedera a ejecutar el script proporcionado en el servidor "localhost\SqIExpress"', 'Continuar(s/n) ? s', 'Se cambio el contexto de la base de datos a \'CalendarioExamenes\'.', '<1 filas afectadas>', '<2 filas afectadas>', '<6 filas afectadas>', '<2 filas afectadas>', 'Script ejecutado con exito en la Base de Datos', and 'Presione una tecla para continuar . . .'. The window has standard Windows window controls (minimize, maximize, close) in the top right corner.

```
C:\Windows\system32\cmd.exe
Bienvenido al script de instalacion de la Base de Datos CalendarioExamenes
Se procedera a ejecutar el script
proporcionado en el servidor "localhost\SqIExpress"
Continuar(s/n) ? s
Se cambio el contexto de la base de datos a 'CalendarioExamenes'.

<1 filas afectadas>

<2 filas afectadas>

<6 filas afectadas>

<2 filas afectadas>
Script ejecutado con exito en la Base de Datos
Presione una tecla para continuar . . .
```

Ilustración 42: Instalación de la base de datos

Pulsaremos cualquier tecla para salir del ejecutable. Ya habremos instalado con éxito la base de datos en nuestro SQL Server, y podremos ejecutar la aplicación de forma correcta.

ANEXO D. Manual de usuario

En el presente anexo se presentará un completo manual de usuario para la aplicación. El objetivo de este manual es guiar al usuario mientras está ejecutando la aplicación, por si tiene alguna duda en algún tipo de funcionalidad.

De esta manera, el usuario podrá consultarlo y despejar las posibles dudas que le vayan surgiendo del funcionamiento de la aplicación.

Se espera que el usuario no tenga que consultarlo apenas, ya que la aplicación es muy intuitiva y además está organizada por pasos secuenciales, por lo que es muy fácil de seguir, y saber qué hacer en cada momento.

Por tanto, el manual se organizará por orden de ejecución, siguiendo los pasos que el usuario debe realizar hasta llegar a generar el calendario de exámenes correctamente y exportarlo a ficheros de salida.

En este manual también se mostrará de forma gráfica las explicaciones para que el usuario sepa exactamente como realizar las funcionalidades y facilitar la comprensión del mismo.

Antes de ejecutar la aplicación no debemos tener en cuenta ningún requisito, solamente los ya vistos en el anexo anterior para poder instalar la aplicación de forma satisfactoria. Por tanto, procederemos a redactar el manual de usuario en la página siguiente, dividido por las distintas pantallas que el usuario se irá encontrando, explicando todas las acciones que el usuario puede realizar en la pantalla y luego explicando cómo realizar cada una.

Archivos adjuntos de la aplicación entregada para la correcta ejecución

Con la solución software se incluyen los siguientes archivos, que son necesarios para el correcto funcionamiento de la aplicación. Los archivos son los siguientes:

- Archivos ejecutables de la aplicación.
- Fichero Excel de entrada, con los datos formateados de las asignaturas:
Este fichero es necesario porque tiene el formato adecuado para poder cargar las asignaturas a la Base de Datos. Lógicamente, este fichero se podrá modificar para cambiar los datos de las asignaturas que se vayan a impartir en cada año, pero conservando el formato original del fichero.

Importante:

Antes de explicar paso por paso la ejecución de la aplicación, el usuario debe saber algo importante: hay funcionalidades que el usuario siempre podrá realizar sin importar en qué paso se encuentra dentro de la ventana principal. Estas funcionalidades son:

- Avanzar al paso siguiente: El usuario siempre podrá avanzar al paso siguiente, si ha cumplido con los campos obligatorios del paso actual. Los campos obligatorios del paso actual vendrán marcados con un asterisco encerrado entre paréntesis, de esta manera: “(*)”

Botón “Avanzar”: 

- Retroceder al paso anterior: El usuario siempre podrá retroceder al paso anterior, por si se equivoca en algún paso y quiere volver atrás, sin importar el paso en el que se encuentre.

Botón “Retroceder”: 

- Salir de la aplicación: El usuario podrá salir de la aplicación si está en la pantalla principal, sin importar en qué paso se encuentre.

Botón “Salir”: 

- Ver información sobre la aplicación: El usuario siempre podrá pulsar el botón “Acerca de” situado en la esquina superior derecha de la pantalla principal para visualizar información acerca de la aplicación.

Botón “Acerca de”: 

Si el usuario pulsa sobre este botón se visualizará la siguiente información:



Ilustración 43: Mensaje de información sobre la aplicación

Además, hay que tener en cuenta otra cosa antes de ejecutar la aplicación. La aplicación necesita que el servicio de la base de datos SqlExpress esté activado. Este servicio debería iniciarse cada vez que se arranca el sistema operativo.

Hay veces, que a causa de un antivirus u otro tipo de software, este servicio no está iniciado, por lo que al ejecutar la aplicación nos encontraremos un mensaje como el siguiente y se nos cerrará la aplicación:

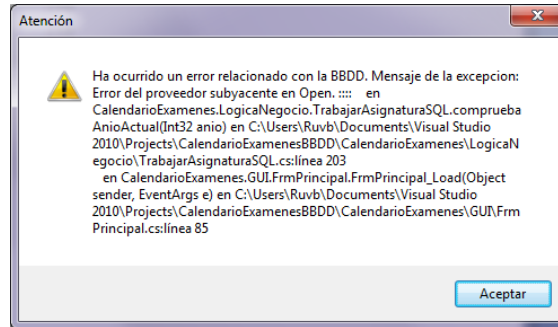


Ilustración 44: Mensaje de aviso indicando que no hay conexión

Si nos ocurre esto, deberemos ir a Inicio -> Panel de Control -> Sistema y seguridad -> Herramientas administrativas -> Servicios. Nos aparecerá una ventana con los servicios del sistema operativo, donde buscaremos el servicio llamado “Sql Server (SQL Express)” y pulsaremos con el botón derecho sobre él, a continuación pulsaremos en “Iniciar” para iniciar el servicio de la base de datos, y así poder ejecutar la aplicación. Aún así, como ya hemos dicho este servicio no debería estar detenido.

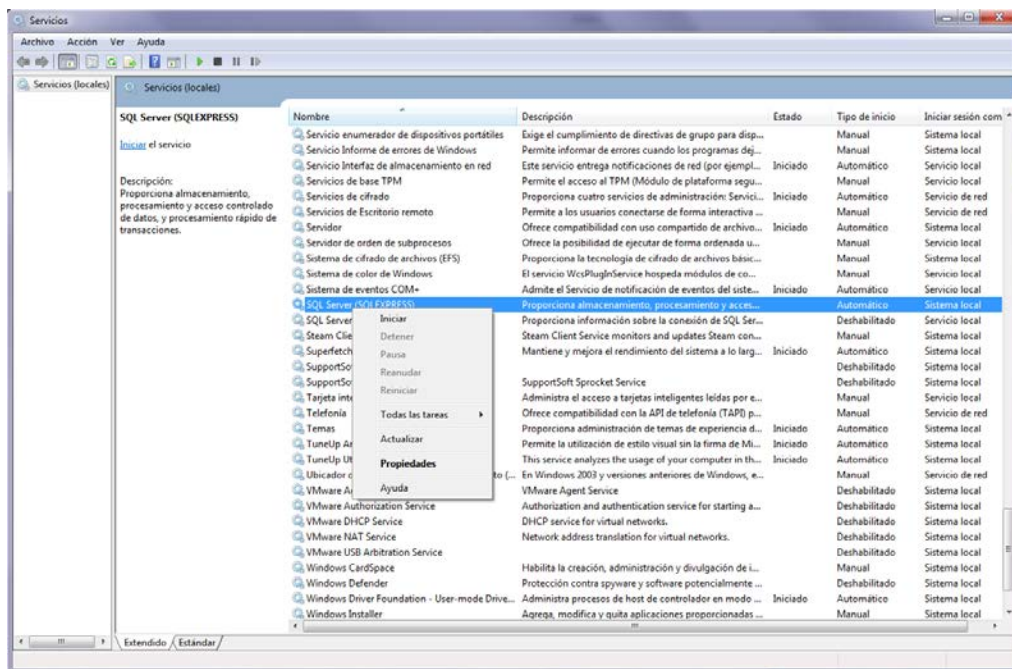


Ilustración 45: Ventana de servicios de Windows

D.1 Primera ejecución

Para ejecutar la aplicación debemos ir a la carpeta donde se encuentran los ejecutables, llamada “CalendarioExámenesEjecutables” y hacer doble click sobre el archivo “CalendarioExámenes.exe”, asegurándonos que el archivo es de tipo “Aplicación”.

Al ejecutarse la aplicación por vez primera, como no habrá datos del año actual de las asignaturas en la base de datos, nos aparecerá el siguiente mensaje:

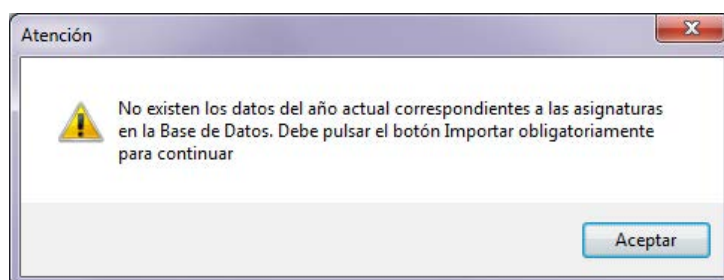


Ilustración 46: Mensaje indicando que no hay datos de asignaturas relativos al año actual

Pulsaremos en el botón “Aceptar”, y nos llevará a la pantalla principal de la aplicación, donde, como no hay datos de asignaturas, estarán todos los controles inhabilitados y solo nos permitirá pulsar en el botón “Importar asignaturas”, para que importemos los datos del fichero Excel de entrada, lo que explicaremos más adelante.

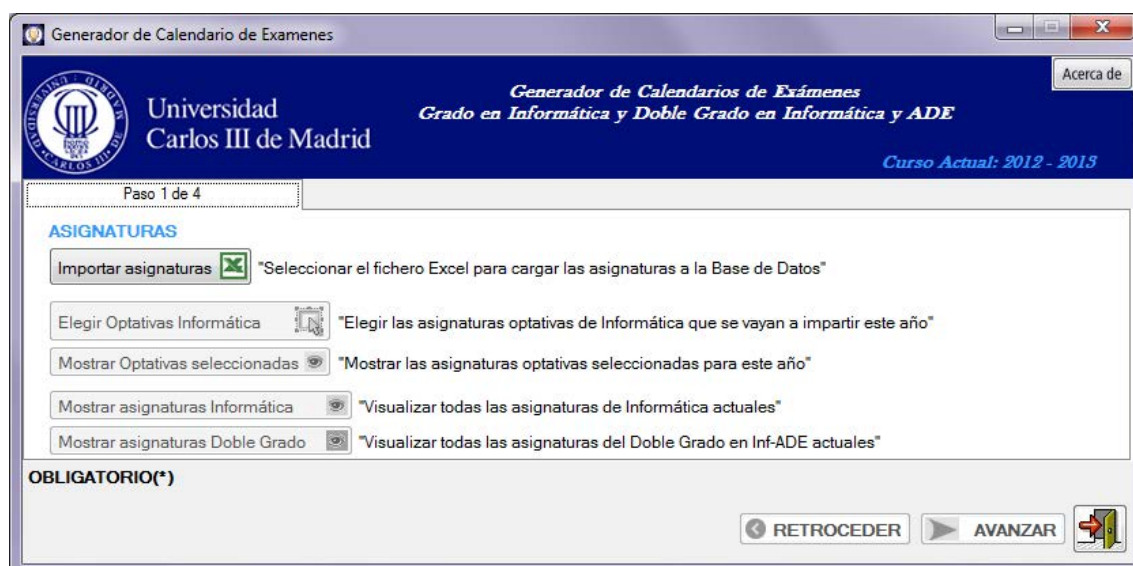


Ilustración 47: Pantalla principal cuando no hay asignaturas (Primer Paso)

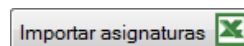
D.2 Primer paso: Asignaturas

En el primer paso tendremos las siguientes funcionalidades:

1. Importar asignaturas de un fichero de entrada Excel a la base de datos.
2. Elegir las optativas de Informática que se vayan a impartir en el curso actual.
3. Visualizar las asignaturas optativas que se han elegido.
4. Visualizar los datos de todas las asignaturas relativas al Grado en Informática que hay en la base de datos, a modo de información.
5. Visualizar los datos de todas las asignaturas relativas al Doble Grado en Informática – ADE que hay en la base de datos, a modo de información.

1. Importar asignaturas de un fichero de entrada Excel a la base de datos:

El usuario podrá importar todos los datos de las asignaturas desde un fichero Excel a la base de datos pulsando en el botón “Importar asignaturas”:



Este botón solo lo pulsaremos cuando no haya datos de las asignaturas del año actual en la base de datos (que nos avisará la aplicación), o cuando queramos importar nuevos datos de asignaturas. Cuando lo pulsemos, puede que nos aparezca el siguiente mensaje pidiendo confirmación, avisándonos de que ya hay datos de las asignaturas en la base de datos, si queremos cambiarlos pulsaremos en “Sí” (La primera vez que se ejecute en cada año académico nos mostrará este mensaje):

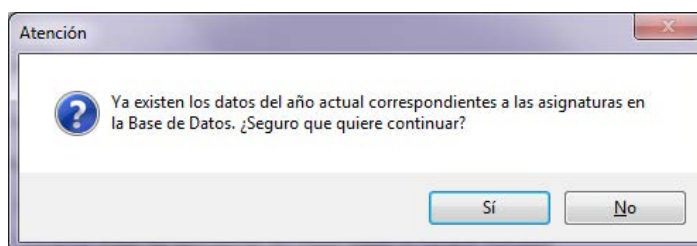


Ilustración 48: Mensaje indicando que ya hay datos

A continuación nos pedirá el fichero Excel de datos, que por defecto se encontrará en la carpeta “Entrada”. Buscaremos el archivo Excel con los datos de las asignaturas y pulsaremos en “Abrir”, como se muestra en la siguiente imagen:

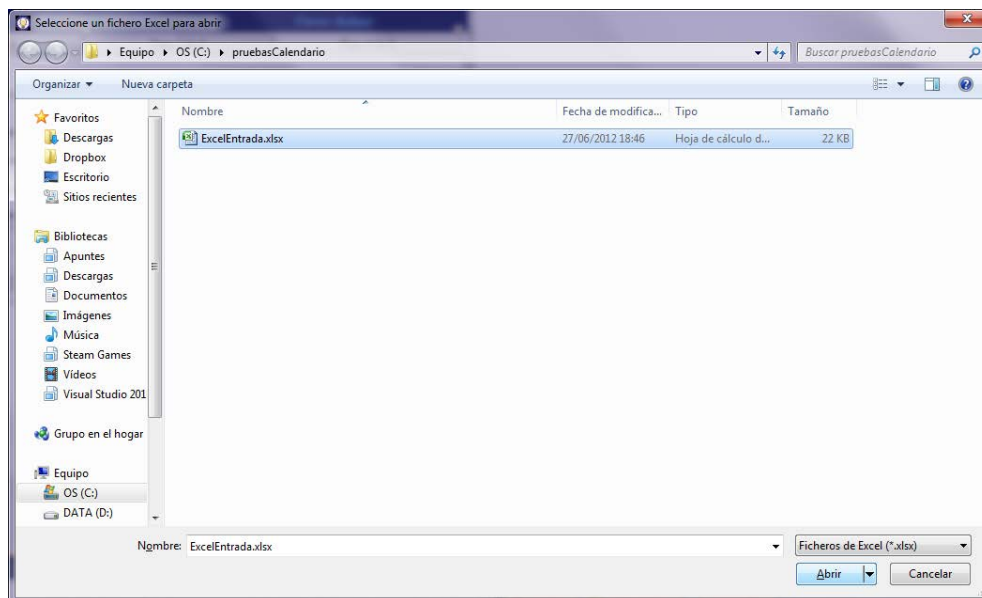


Ilustración 49: Cuadro de diálogo para abrir nuestro fichero Excel

Cuando hayamos hecho esto nos aparecerá un mensaje indicándonos que se están importando los datos a la base de datos y que esperemos, esto no debería durar más de cinco segundos:

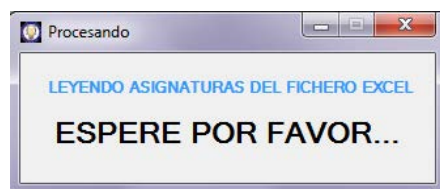


Ilustración 50: Mensaje de espera

Cuando se haya completado el proceso, se mostrará un mensaje satisfactorio al usuario, indicándonos que el fichero se ha leído correctamente, como el siguiente:

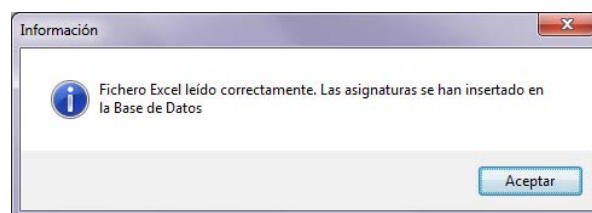


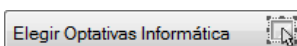
Ilustración 51: Mensaje de éxito al leer fichero de entrada

Si abrimos el fichero Excel de entrada para visualizarlo o modificarlo para añadir/quitar asignaturas para el curso actual, nos fijaremos que hay dos asignaturas con un color de fuente rojo, esto es para remarcarlas ya que estas dos asignaturas son dos asignaturas “especiales”. Son dos asignaturas compartidas por las dos titulaciones que deberían tener el examen el mismo día, pero no es así, ya que no comparten grupo. Por eso en el fichero Excel de entrada vienen con un código distinto en el Doble Grado y en el Grado de Informática, pero es normal.

2. Elegir las optativas de Informática que se vayan a impartir el curso actual:

Si lo desea, el usuario podrá elegir entre el conjunto de asignaturas optativas pertenecientes a la titulación Grado en Ingeniería Informática aquellas que se vayan a impartir en el curso actual. Si quiere que se seleccionen todas para la generación del calendario no es necesario que se pulse en este botón.

Botón “Elegir optativas Informática”:



Al pulsar en este botón aparecerá la siguiente pantalla:

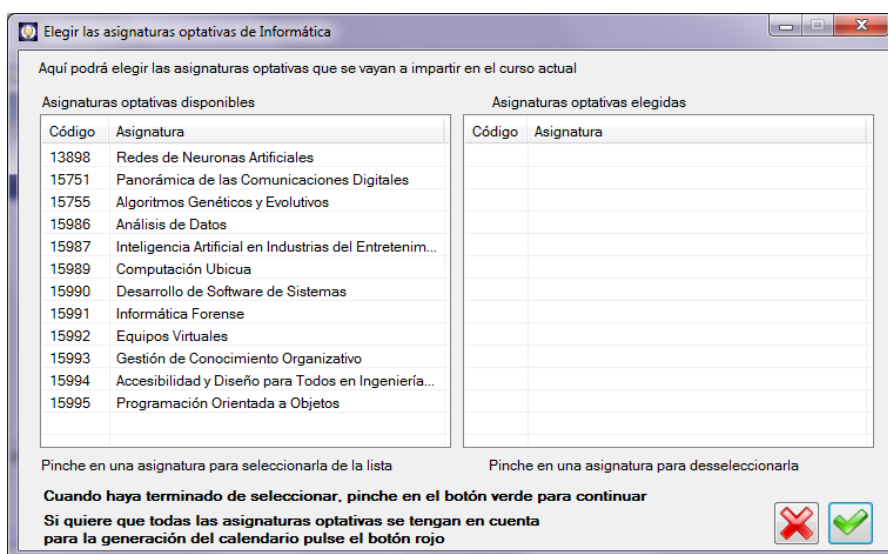


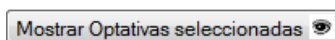
Ilustración 52: Pantalla de elección de optativas

En esta pantalla podremos seleccionar optativas del lado izquierdo simplemente con pinchar en el nombre de la asignatura, e irán apareciendo las que vayamos seleccionando en el lado derecho. Cuando estemos listos, pulsaremos en el botón del “tick” verde. Si en realidad queremos seleccionar todas y salir de esta pantalla pulsaremos en el botón rojo a la izquierda.

3. Visualizar las asignaturas optativas que se han elegido:

El usuario también podrá ver las asignaturas optativas de Informática que ha seleccionado para la generación del calendario de exámenes pulsando en el botón “Mostrar optativas seleccionadas”.

Botón “Mostrar optativas seleccionadas”:



Al pulsar este botón se mostrarán las asignaturas optativas que el usuario haya seleccionado anteriormente. Si no había seleccionado ninguna, se mostrarán todas las asignaturas optativas de Informática, las cuales se tendrán en cuenta para la generación del calendario de exámenes. Nos aparecerá la siguiente pantalla:

Curso	Cuatrimestre	Código	Nombre	Especialidad	Optativa
4	1	15990	Desarrollo de Softw...	INGENIERIA DE CO...	SI
4	1	15751	Panorámica de las ...	INGENIERIA DE CO...	SI
4	1	15986	Análisis de Datos	COMPUTACION	SI
4	1	15987	Inteligencia Artificial ...	COMPUTACION	SI
4	1	15995	Programación Orien...	COMPUTACION	SI
4	1	15991	Informática Forense	INGENIERIA DE CO...	SI
4	1	15992	Equipos Virtuales	SISTEMAS DE LA I...	SI
4	1	13898	Redes de Neuronas...	COMPUTACION - SI...	SI
4	1	15994	Accesibilidad y Dise...	SISTEMAS DE LA I...	SI

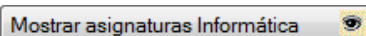
Asignaturas mostradas: 9 asignaturas
 Esta ventana muestra una lista específica de asignaturas.
 Si lo desea, puede ordenar las asignaturas por criterio pinchando en el nombre de la columna

Ilustración 53: Pantalla de visualización de optativas

Esta pantalla la podremos maximizar y redimensionar a nuestro antojo para visualizar bien todos los datos de las asignaturas. Cuando las hayamos visto podremos pulsar en el botón de la puerta para salir de esta pantalla.

4. Visualizar los datos de todas las asignaturas relativas al Grado en Informática que hay en la base de datos, a modo de información:

Botón “Mostrar asignaturas Informática”:



Pinchando en este botón, se mostrará una pantalla al usuario donde se visualizarán los datos de todas las asignaturas del año actual pertenecientes a la titulación Grado en Informática que hay en la base de datos.

De nuevo, esta ventana podremos maximizarla o redimensionarla como queramos para visualizar bien todos los datos, y cuando queramos cerrarla pulsaremos en el botón de la puerta.

Asignaturas mostradas: INFORMÁTICA

	Curso	Cuatrimestre	Código	Nombre	Especialidad	Optativa
▶	1	1	13870	Álgebra Lineal	COMUN	NO
	1	1	13866	Cálculo	COMUN	NO
	1	1	13867	Física	COMUN	NO
	1	1	13868	Programación	COMUN	NO
	1	2	13869	Tecnología de Com...	COMUN	NO
	1	2	13871	Principios Físicos d...	COMUN	NO
	1	2	13873	Estructura de Datos...	COMUN	NO
	1	2	15970	Lógica	COMUN	NO
	1	2	15971	Matemática Discreta	COMUN	NO
	2	1	15975	Cálculo Diferencial ...	COMUN	NO
	2	1	13874	Estructura de Comp...	COMUN	NO
	2	1	13875	Fundamentos de G...	COMUN	NO
	2	1	13876	Estadística	COMUN	NO
	2	1	13877	Teoría de Autómata...	COMUN	NO
	2	1	13878	Sistemas Operativos	COMUN	NO
	2	2	13881	Ficheros y Bases de...	COMUN	NO
	2	2	13883	Inteligencia Artificial	COMUN	NO
	2	2	15972	Principios del Desa...	COMUN	NO
	2	2	15973	Criptografía y Segur...	COMUN	NO
	3	1	15974	Ingeniería del Softw...	COMUN	NO
	3	1	15976	Heurística y Optimiz...	COMUN	NO

Asignaturas mostradas: 58 asignaturas

Esta ventana muestra una lista específica de asignaturas.
Si lo desea, puede ordenar las asignaturas por criterio pinchando en el nombre de la columna

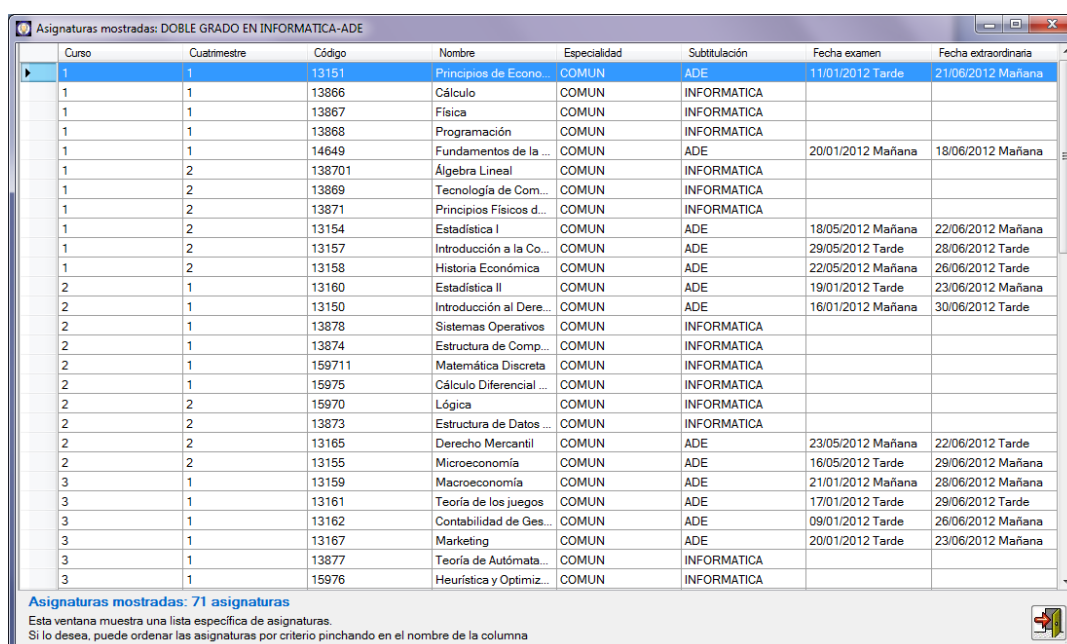
Ilustración 54: Pantalla de visualización de asignaturas del Grado en Informática

5. Visualizar los datos de todas las asignaturas relativas al Doble Grado en Informática – ADE que hay en la base de datos, a modo de información:

Botón “Mostrar asignaturas Doble Grado”: 

Pinchando en este botón, se mostrará una pantalla al usuario donde se visualizarán los datos de todas las asignaturas del año actual pertenecientes a la titulación Doble Grado en Informática – ADE que hay en la base de datos.

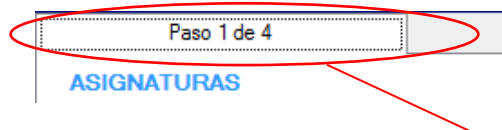
De nuevo, esta ventana podremos maximizarla o redimensionarla como queramos para visualizar bien todos los datos, y cuando queramos cerrarla pulsaremos en el botón de la puerta.



Curso	Cuatrimestre	Código	Nombre	Especialidad	Subtitulación	Fecha examen	Fecha extraordinaria
1	1	13151	Principios de Econo...	COMUN	ADE	11/01/2012 Tarde	21/06/2012 Mañana
1	1	13866	Cálculo	COMUN	INFORMATICA		
1	1	13867	Física	COMUN	INFORMATICA		
1	1	13868	Programación	COMUN	INFORMATICA		
1	1	14649	Fundamentos de la ...	COMUN	ADE	20/01/2012 Mañana	18/06/2012 Mañana
1	2	138701	Álgebra Lineal	COMUN	INFORMATICA		
1	2	13869	Tecnología de Com...	COMUN	INFORMATICA		
1	2	13871	Principios Físicos d...	COMUN	INFORMATICA		
1	2	13154	Estadística I	COMUN	ADE	18/05/2012 Mañana	22/06/2012 Mañana
1	2	13157	Introducción a la Co...	COMUN	ADE	29/05/2012 Tarde	28/06/2012 Tarde
1	2	13158	Historia Económica	COMUN	ADE	22/05/2012 Mañana	26/06/2012 Tarde
2	1	13160	Estadística II	COMUN	ADE	19/01/2012 Tarde	23/06/2012 Mañana
2	1	13150	Introducción al Dere...	COMUN	ADE	16/01/2012 Mañana	30/06/2012 Tarde
2	1	13878	Sistemas Operativos	COMUN	INFORMATICA		
2	1	13874	Estructura de Comp...	COMUN	INFORMATICA		
2	1	159711	Matemática Discreta	COMUN	INFORMATICA		
2	1	15975	Cálculo Diferencial ...	COMUN	INFORMATICA		
2	2	15970	Lógica	COMUN	INFORMATICA		
2	2	13873	Estructura de Datos ...	COMUN	INFORMATICA		
2	2	13165	Derecho Mercantil	COMUN	ADE	23/05/2012 Mañana	22/06/2012 Tarde
2	2	13155	Microeconomía	COMUN	ADE	16/05/2012 Tarde	29/06/2012 Mañana
3	1	13159	Macroeconomía	COMUN	ADE	21/01/2012 Mañana	28/06/2012 Mañana
3	1	13161	Teoría de los juegos	COMUN	ADE	17/01/2012 Tarde	29/06/2012 Tarde
3	1	13162	Contabilidad de Ges...	COMUN	ADE	09/01/2012 Tarde	26/06/2012 Mañana
3	1	13167	Marketing	COMUN	ADE	20/01/2012 Tarde	23/06/2012 Mañana
3	1	13877	Teoría de Automata...	COMUN	INFORMATICA		
3	1	15976	Heurística y Optimiz...	COMUN	INFORMATICA		

Ilustración 55: Pantalla de visualización de asignaturas del Doble Grado

Ya hemos visto todas las acciones que el usuario puede realizar en el primer paso de la aplicación. Si pulsa en el botón “Avanzar” se irá al paso segundo de la aplicación.



Esto nos indica en qué paso nos encontramos

D.3 Segundo paso: Fechas de exámenes

Al haber pulsado el botón “Avanzar” en el primer paso, se nos mostrará el segundo paso, en el que tendremos que elegir el período deseado para generar el calendario de exámenes y las fechas de inicio y de fin del calendario.

Ilustración 56: Pantalla principal de la aplicación (Segundo paso)

En este paso el usuario podrá realizar las siguientes acciones:

- Seleccionar cuatrimestre: Podremos pinchar en “Primer Cuatrimestre”, “Segundo Cuatrimestre” o “Convocatoria Extraordinaria”, para generar el calendario de exámenes para uno de los tres períodos. Esto es obligatorio para poder avanzar al paso siguiente.

Ilustración 57: Selección del período

- Al haber seleccionado un período nos aparecerán los campos para que rellenemos la fecha de inicio y de fin del calendario, lo cual será también obligatorio, y permanecerán con fondo rojo hasta que no hayamos seleccionado las fechas, momento en el cual tendrán fondo verde, momento en el cual ya podremos pasar al paso siguiente.

The image shows a user interface for selecting exam dates. It consists of two rows of input fields. The top row shows empty fields with red backgrounds, labeled 'Inicio Exámenes 2º Cuatrimestre (*)' and 'Fin Exámenes 2º Cuatrimestre (*)'. The bottom row shows the same fields with dates entered: '14/05/2012' and '28/05/2012', with green backgrounds. Each field has a small calendar icon next to it.

Ilustración 58: Selección de fechas de inicio y fin del período

- No contar con sábados: El usuario podrá elegir si quiere que los sábados no se tengan en cuenta para colocar exámenes en el calendario a generar. Esto es opcional para poder pasar al siguiente paso. Si quiere que los sábados no pueda haber exámenes simplemente tendrá que marcar la opción.

☐ No contar con los sábados

Ilustración 59: No contar con los sábados

D.4 Tercer paso: Algoritmo a utilizar

Una vez hayamos pulsado en el botón “Aceptar” en el segundo paso habiendo introducido los campos obligatorios, se nos mostrará la pantalla del tercer paso. Aquí el usuario deberá seleccionar un algoritmo para la generación del calendario.

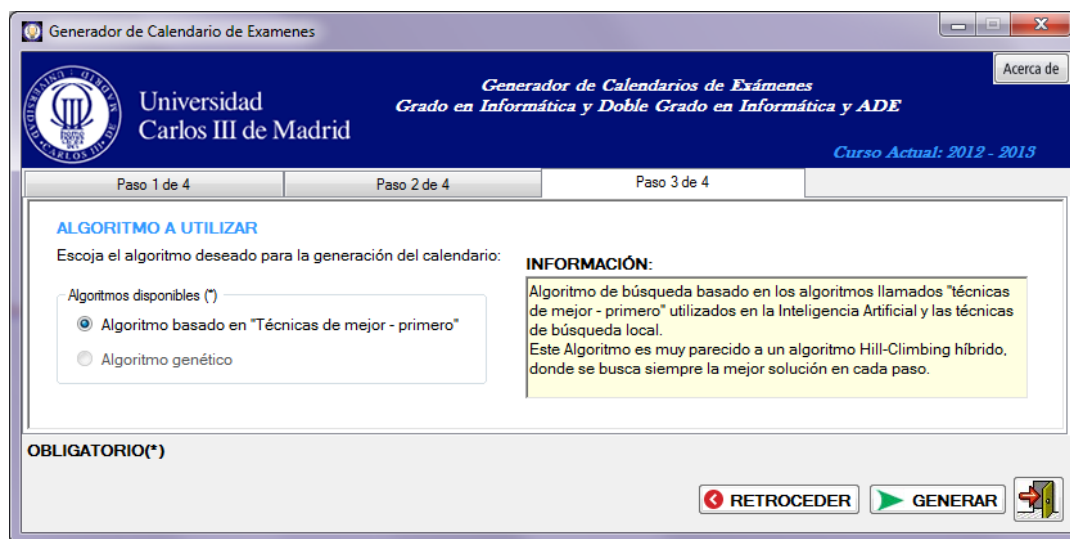


Ilustración 60: Pantalla principal (Tercer Paso)

Aquí el usuario podrá realizar las siguientes acciones:

- Seleccionar un algoritmo: Esto es obligatorio para poder generar el calendario de exámenes. Al pinchar en un algoritmo se mostrará un texto a la izquierda describiendo brevemente el algoritmo.

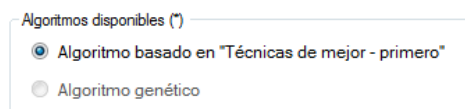



Ilustración 61: Selección de algoritmo

Una vez hayamos seleccionado un algoritmo pulsaremos en el botón “Generar”, para que empiece el proceso de generación del calendario de exámenes.

- En este momento nos aparecerá una barra de progreso que nos indica cómo va la generación y lo único que se podrá hacer en el momento de la generación es cancelar el proceso, todos los demás controles estarán inhabilitados.

Botón Cancelar: 

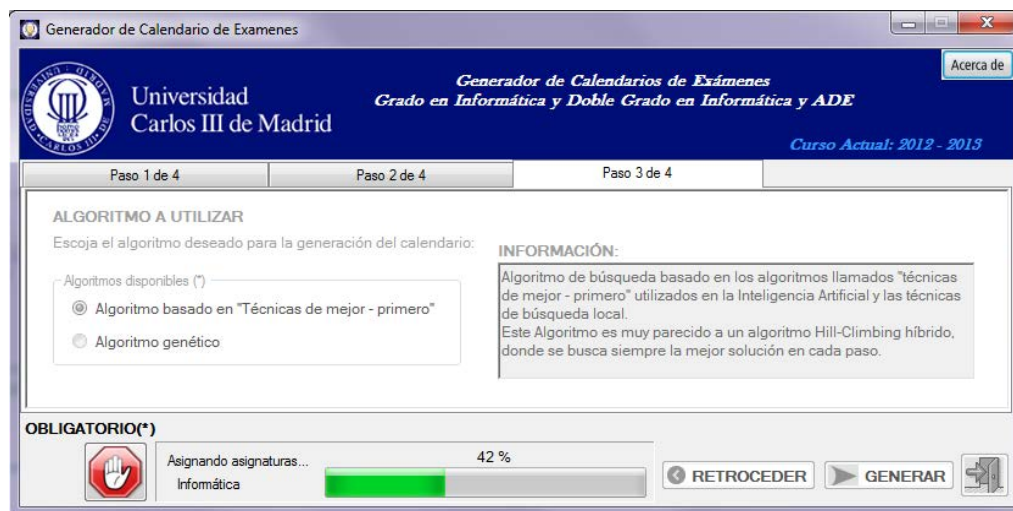


Ilustración 62: Pantalla principal durante el proceso de generación

Si pulsamos en el botón “Cancelar” se detendrá el proceso de generación y volveremos a la pantalla del tercer paso.

Una vez se haya completado el proceso de generación del calendario nos mostrará un mensaje, como el siguiente:

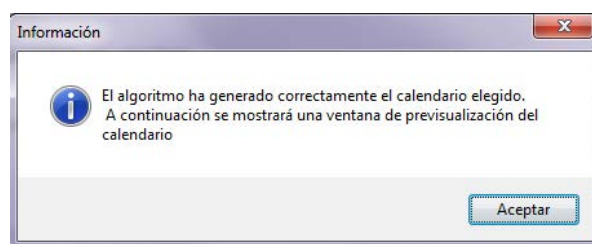


Ilustración 63: Mensaje de terminación de la generación del calendario

A continuación, se nos mostrará una ventana de previsualización del calendario de exámenes generado, donde podremos ver las fechas de los exámenes de las asignaturas correspondientes al período que hayamos elegido anteriormente, como se muestra a continuación:

Previsualización del calendario: Segundo cuatrimestre

Titulación	Curso	Código	Nombre	Especialidad	Fecha
Doble Grado Inf-ADE	2	13873	Estructura de Datos y Al...	COMUN	12/05/2012 Mañana
Doble Grado Inf-ADE	4	13893	Ingeniería de la Segurid...	INGENIERIA DE COMP...	12/05/2012 Mañana
Doble Grado Inf-ADE	4	15763	Teoría Avanzada de la ...	COMPUTACION	12/05/2012 Mañana
Informática	1	15971	Matemática Discreta	COMUN	21/05/2012 Mañana
Doble Grado Inf-ADE	4	13885	Organización de Comp...	INGENIERIA DE COMP...	10/05/2012 Mañana
Doble Grado Inf-ADE	5	15758	Informática Gráfica	COMPUTACION - INGE...	19/05/2012 Mañana
Doble Grado Inf-ADE	5	13896	Métodos y Técnicas de ...	SISTEMAS DE LA INFO...	19/05/2012 Mañana
Doble Grado Inf-ADE	1	13869	Tecnología de Computa...	COMUN	09/05/2012 Mañana
Doble Grado Inf-ADE	3	15973	Criptografía y Seguridad...	COMUN	09/05/2012 Mañana
Doble Grado Inf-ADE	5	15754	Sistemas Distribuidos	INGENIERIA DE COMP...	09/05/2012 Mañana
Doble Grado Inf-ADE	5	13890	Procesadores del Leng...	COMPUTACION	09/05/2012 Mañana
Doble Grado Inf-ADE	5	15812	Técnicas Ágiles del Des...	SISTEMAS DE LA INFO...	09/05/2012 Mañana
Doble Grado Inf-ADE	1	13871	Principios Físicos de la I...	COMUN	15/05/2012 Mañana
Doble Grado Inf-ADE	3	13881	Ficheros y Bases de Dat...	COMUN	15/05/2012 Mañana
Doble Grado Inf-ADE	2	15970	Lógica	COMUN	07/05/2012 Mañana
Doble Grado Inf-ADE	4	13889	Diseño de Sistemas Op...	COMUN	07/05/2012 Mañana
Doble Grado Inf-ADE	4	13892	Dirección de Proyectos ...	COMUN	17/05/2012 Mañana
Doble Grado Inf-ADE	5	13891	Diseño de Sistemas Int...	COMPUTACION	23/05/2012 Mañana
Doble Grado Inf-ADE	5	15809	Diseño y Administración...	SISTEMAS DE LA INFO...	23/05/2012 Mañana
Doble Grado Inf-ADE	4	13883	Inteligencia Artificial	COMUN	14/05/2012 Mañana
Doble Grado Inf-ADE	1	138701	Álgebra Lineal	COMUN	11/05/2012 Mañana
Doble Grado Inf-ADE	3	15972	Principios del Desarrollo ...	COMUN	11/05/2012 Mañana
Doble Grado Inf-ADE	5	15760	Recuperación y Acceso ...	SISTEMAS DE LA INFO...	11/05/2012 Mañana
Doble Grado Inf-ADE	5	15761	Seguridad en Dispositiv...	INGENIERIA DE COMP...	11/05/2012 Mañana
Doble Grado Inf-ADE	5	15757	Aprendizaje Automático	COMPUTACION	11/05/2012 Mañana

En esta ventana se muestran las asignaturas que ha colocado el calendario y sus respectivas fechas de examen.
Si lo desea, puede ordenar las asignaturas por criterio pinchando en el nombre de la columna.

Asignaturas generadas en el calendario: 25 asignaturas

¿Le gusta este resultado?

Ilustración 64: Ventana de previsualización del calendario

Esta ventana podremos maximizarla y redimensionarla a nuestro antojo para poder ver bien todas las asignaturas y sus respectivas fechas de exámenes. También se podrá ordenar por columnas, pinchando en los encabezados de cada una.

El usuario no debe asustarse si no ve suficientes asignaturas relativas a una titulación, ya que en esta ventana no se muestran las asignaturas por duplicado si son compartidas por las dos titulaciones.

Si creemos que este resultado es aceptable pulsaremos en el botón “Sí” (mano verde), y nos llevará al paso siguiente y último de la aplicación. Si pulsamos en el botón “No... Probaré otra vez” (mano roja), volveremos de nuevo al tercer paso de la aplicación.

D.5 Cuarto paso: Exportar resultados

Este es el último paso de la aplicación, aquí podremos exportar el resultado como distintos calendarios por titulación y por curso, para que sea más fácil su lectura y seguimiento.

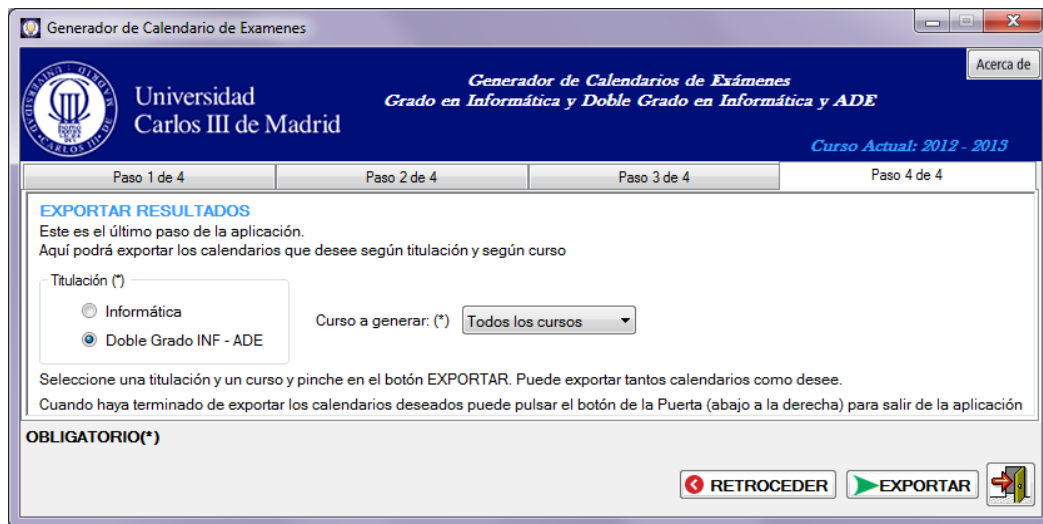


Ilustración 65: Pantalla principal (Cuarto paso)

Las acciones que el usuario puede llevar a cabo en este paso son las siguientes:

- Exportar el resultado en distintos calendarios por titulación y por curso: El usuario deberá pinchar en una titulación, y posteriormente elegir un curso de la lista desplegable de la derecha. Esto es obligatorio para poder exportar un calendario.

A continuación pulsaremos en el botón “Exportar”, y la aplicación exportará un calendario en formato Excel para que el usuario pueda visualizarlo mejor, cuando termine nos mostrará el siguiente mensaje, indicándonos que el calendario se ha exportado a una carpeta llamada “CalendariosGenerados”:

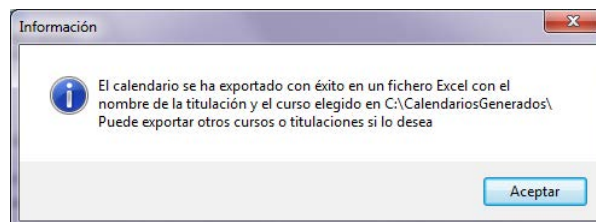


Ilustración 66: Mensaje de éxito al exportar

Podremos exportar los calendarios que queramos seleccionando de nuevo una titulación y un curso, y volviendo a pinchar en el botón “Exportar”. Cuando hayamos terminado de exportar los calendarios deseados, podremos pinchar en el botón de “Salir” (icono de la puerta), para salir de la aplicación.

También podremos pinchar en el botón “Retroceder”, por si no queremos salir aún y queremos seleccionar un período distinto a generar, por ejemplo.

Aquí finaliza el manual de usuario de la aplicación, en el que pretendemos haber cubierto todas las funcionalidades de la aplicación.